Oracle® Database

2 Day DBA

11*g* Release 1 (11.1)

Part No. B28301-02

September 2007

ORACLE®

Oracle Database 2 Day DBA 11*g* Release 1 (11.1)

Part No. B28301-02

Primary Author: Steve Fogel, Janet Stern

Contributing Author: Lance Ashdown, David Austin, Craig Foch, Sushil Kumar, Roza Leyderman, Colin McGregor, Antonio Romero, and Douglas Williams

Contributor: Bharat Baddepudi, Prasad Bagal, Timothy Chien, Benoit Dageville, Suddip Datta, Kakali Das, Mark Dilman, Prabhaker Gongloor, Shivani Gupta, Balaji Krishnan, Vasudha Krishnaswamy, Rich Long, Venkat Maddali, Matthew McKerley, Gaurav Manglik, Ed Miner, Mughees Minhas, Dheeraj Pandey, Kant Patel, Hanlin Qian, Ravi Rajamani, Bipul Sinha, Mark Townsend, Xiaofang Wang, and Mike Zampiceni

Contents

Preface xiii

Audience xiii
Documentation Accessibility xiii
Related Documentation xiv
Conventions xv

1 Introduction

About This Guide 1-1
What This Guide Is Not 1-1
How to Use This Guide with Related Material 1-1
About Oracle Database 1-2
Common Oracle DBA Tasks 1-2
Tools for Administering the Database 1-3

2 Installing Oracle Database and Creating a Database

Overview of Installing Oracle Database Software and Creating a Database 2-1
Checking Prerequisites 2-2
Installation Choices 2-2
Basic Installation 2-2
Advanced Installation 2-3
Installing Oracle Database Software 2-5
Using DBCA to Create and Configure a Database 2-8
Starting DBCA 2-9
Creating a Database Using DBCA 2-9
Step 2 - Database Templates 2-10
Step 3 - Database Identification 2-10
Step 4 - Management Options 2-10
Step 5 - Database Credentials 2-10
Step 6 - Storage Options 2-10
Step 7 - Database File Locations 2-11
Step 8 - Recovery Configuration 2-11
Step 9 - Database Content 2-12
Step 10 - Initialization Parameters 2-12
Step 11 - Security Settings 2-15
Step 12 - Automatic Maintenance Tasks 2-15

Step 13 - Database Storage 2-15
Step 14 - Creation Options 2-15
Changing the Configuration of a Database Using DBCA 2-15
Deleting a Database Using DBCA 2-15
Managing Templates with DBCA 2-15
Advantages of Using Templates 2-16
Types of Templates 2-16
DBCA Templates Provided by Oracle Database 2-16
Creating Templates Using DBCA 2-17
Deleting DBCA Templates 2-18
Configuring Automatic Storage Management Using DBCA 2-18
Installation: Oracle By Example Series 2-18

3 Getting Started with Database Administration

Managing Your Database: A Roadmap 3-1
Introduction to Oracle Enterprise Manager Database Control 3-2
Database Management Features 3-2
Database Home Page 3-2
Online Help 3-3
Navigational Features 3-3
Starting and Stopping the Database Control Console Process 3-3
Accessing the Database Home Page 3-4
Access to Database Control for Nonadministrative Users 3-6
Creating Database Control Administrative Users 3-6
Setting Database Control Preferences 3-8
About Database Control Preferences 3-8
Defining Blackout Periods 3-8
Setting Preferred Credentials 3-10
About SQL and SQL-Based Management Tools 3-11
Starting SQL*Plus and Connecting to the Database 3-11
About SQL Developer 3-13
Getting Started with Oracle Enterprise Manager: Oracle By Example Series 3-13

4 Configuring the Network Environment

Understanding Network Configuration 4-1
Oracle Net Listener Configuration 4-1
Connect Descriptors 4-2
Connection Requests 4-2
Naming Methods 4-3
Easy Connect Naming 4-3
Local Naming 4-3
Directory Naming 4-3
Tools for Network Configuration 4-3
Net Configuration Assistant 4-3
Oracle Enterprise Manager 4-4
Oracle Net Manager 4-4
Viewing Listener Configuration 4-4

Starting and Stopping the Listener ... 4-5
Connecting to an Oracle Database from a Client Computer ... 4-6
Networking: Oracle By Example Series ... 4-7

5 Managing the Oracle Instance

Overview of the Oracle Instance and Instance Management ... 5-1
About Initialization Parameters ... 5-2
About Background Processes ... 5-3
About Server and Client Processes ... 5-4
About Instance Memory Structure ... 5-4
System Global Area ... 5-5
Program Global Area ... 5-5
About Instance Startup and Shutdown ... 5-6
About Administration Privileges for Startup and Shutdown ... 5-6
About Instance Startup ... 5-6
About Instance Shutdown ... 5-7
Shutting Down and Starting Up the Oracle Instance ... 5-7
Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control ... 5-8
Shutting Down and Starting Up Using the Windows Services Program ... 5-9
Viewing and Modifying Initialization Parameters ... 5-10
Managing Memory ... 5-12
About Memory Management ... 5-13
Enabling Automatic Memory Management ... 5-15
Modifying Memory Settings – Automatic Memory Management ... 5-16
Modifying Memory Settings – Automatic Shared Memory Management ... 5-18
Modifying Memory Settings – Automatic PGA Memory Management ... 5-20
Instances: Oracle By Example Series ... 5-21

6 Managing Database Storage Structures

About Database Storage Structures ... 6-1
About Control Files ... 6-3
About Online Redo Log Files ... 6-3
About Archived Redo Log Files ... 6-4
About Rollback Segments ... 6-5
About Datafiles ... 6-5
About Tablespaces ... 6-6
Locally Managed Tablespaces Compared to Dictionary-Managed Tablespaces ... 6-7
Tablespace Type ... 6-7
Tablespace Status ... 6-8
Autoextend Tablespace ... 6-8
About Other Storage Structures ... 6-9
Initialization Parameter File ... 6-9
Password File ... 6-9
Backup Files ... 6-10
Viewing Database Storage Structure Information ... 6-10
Viewing Control File Information ... 6-10

Viewing Online Redo Log File Information ... 6-10
Viewing Archive Log Information ... 6-11
Viewing Datafile Information ... 6-11
Viewing Tablespace Information ... 6-11
Performing Common Database Storage Tasks ... 6-12
Creating a Tablespace ... 6-12
Modifying a Tablespace ... 6-15
Setting a Tablespace to Automatically Extend ... 6-15
Changing Space Usage Alert Thresholds for a Tablespace ... 6-16
Taking a Tablespace Offline ... 6-17
Dropping a Tablespace ... 6-18
Reclaiming Unused Space ... 6-19
About Reclaimable Unused Space ... 6-19
About the Segment Advisor ... 6-19
Viewing Segment Advisor Recommendations and Reclaiming Space ... 6-20
Running the Segment Advisor Manually ... 6-23
Managing the Redo Log ... 6-23
Multiplexing the Redo Log ... 6-23
Switching a Log File ... 6-24
Managing Undo Data ... 6-25
About Undo Data ... 6-25
About Managing Undo Data ... 6-26
Viewing Undo Information ... 6-27
Computing the Minimum Undo Tablespace Size Using the Undo Advisor ... 6-28
Changing the Undo Tablespace to a Fixed Size ... 6-30
Storage: Oracle By Example Series ... 6-31

7 Administering User Accounts and Security

About User Accounts ... 7-1
About User Privileges and Roles ... 7-3
About Administrative Accounts and Privileges ... 7-4
SYS and SYSTEM Users ... 7-4
SYSDBA and SYSOPER System Privileges ... 7-5
Administering Roles ... 7-5
Viewing Roles ... 7-6
Example: Creating a Role ... 7-7
Example: Modifying a Role ... 7-8
Deleting a Role ... 7-9
Administering Database User Accounts ... 7-9
Viewing User Accounts ... 7-10
Example: Creating a User Account ... 7-11
Creating a New User Account by Duplicating an Existing User Account ... 7-12
Example: Granting Privileges and Roles to a User Account ... 7-13
Example: Assigning a Tablespace Quota to a User Account ... 7-15
Example: Modifying a User Account ... 7-16
Locking and Unlocking User Accounts ... 7-17
Expiring a User Password ... 7-17

Example: Deleting a User Account ... 7-18
Setting the Database Password Policy ... 7-18
About Password Policies ... 7-19
Modifying the Default Password Policy ... 7-19
Users: Oracle By Example Series ... 7-20

8 Managing Schema Objects

About Schema Objects ... 8-1
About Schema Object Management Privileges ... 8-2
Managing Tables ... 8-2
About Tables ... 8-3
About Table Types ... 8-3
About Table Column Attributes ... 8-3
About Table-Level Constraints ... 8-5
About Table Storage Attributes ... 8-6
Other Table Creation Considerations ... 8-6
Viewing Tables ... 8-7
Viewing Table Data ... 8-8
Example: Creating a Table ... 8-9
Modifying Table Attributes ... 8-12
Example: Adding Table Columns ... 8-12
Example: Deleting a Table Column ... 8-13
Example: Adding a New Table Constraint ... 8-13
Example: Modifying an Existing Table Constraint ... 8-15
Example: Deleting a Table Constraint ... 8-15
Example: Loading Data into a Table ... 8-16
Deleting a Table ... 8-18
Managing Indexes ... 8-18
About Indexes ... 8-18
Indexes and Performance ... 8-19
Index Attributes ... 8-19
Viewing Indexes ... 8-20
Example: Creating an Index ... 8-21
Example: Deleting an Index ... 8-22
Managing Views ... 8-23
About Views ... 8-23
Displaying Views ... 8-24
Example: Creating a View ... 8-25
Example: Deleting a View ... 8-26
Managing Program Code Stored in the Database ... 8-26
About Program Code Stored in the Database ... 8-27
Validating (Compiling) Invalid Schema Objects ... 8-27
Working with Other Schema Objects ... 8-29
Schemas: Oracle By Example Series ... 8-30

9 Performing Backup and Recovery

Overview of Database Backup and Recovery ... 9-1
Database Backup and Recovery Concepts ... 9-2
Consistent and Inconsistent Backups ... 9-2
Media Recovery ... 9-3
Flash Recovery Area ... 9-3
RMAN Repository ... 9-4
Configuring Your Database for Basic Backup and Recovery ... 9-4
Specifying Credentials for Backup and Recovery Using Database Control ... 9-4
Planning Space Usage and Location for the Flash Recovery Area ... 9-5
About the Backup Retention Policy and the Flash Recovery Area ... 9-5
About the Flash Recovery Area Size ... 9-6
Configuring Recovery Settings ... 9-6
Monitoring Flash Recovery Area Usage ... 9-7
Configuring Backup Settings ... 9-7
About RMAN Backups ... 9-7
Configuring Backup Device Settings ... 9-8
Configuring Backup Policy Settings ... 9-9
Backing Up Your Database ... 9-10
RMAN Backup Concepts ... 9-10
Full Backups of Datafiles ... 9-10
Incremental Backups of Datafiles ... 9-10
Incrementally Updated Backups: Rolling Forward Image Copies of Datafiles ... 9-11
Backup Tags ... 9-11
Performing and Scheduling Backups Using Database Control ... 9-11
Performing a Whole Database Backup ... 9-11
Using the Oracle Suggested Backup Strategy ... 9-14
Scheduling Miscellaneous Backup Tasks ... 9-15
Validating Backups and Testing Your Backup Strategy ... 9-16
Validating Selected Backups ... 9-16
Validating Backups for Restore Operations ... 9-17
Displaying Backup Reports ... 9-17
Managing Backups ... 9-18
About Backup Management ... 9-18
Cross-Checking Backups ... 9-20
Deleting Expired Backups ... 9-21
Marking Backups as Available or Unavailable ... 9-21
Deleting Obsolete Backups ... 9-22
Performing Oracle Advised Recovery ... 9-22
About Data Recovery Advisor ... 9-23
Using Data Recovery Advisor ... 9-23
Performing User-Directed Recovery ... 9-25
Rewinding a Table Using Oracle Flashback Table ... 9-26
Enabling Row Movement on a Table ... 9-26
Performing a Flashback Table Operation ... 9-27
Recovering a Dropped Table Using Oracle Flashback Drop ... 9-28
Dropping a Table ... 9-28

Retrieving a Dropped Table ... 9-29
Rewinding a Database Using Oracle Flashback Database ... 9-30
Restoring and Recovering the Database ... 9-31
Backup and Recovery: Oracle By Example Series ... 9-32

10 Monitoring and Tuning the Database

Proactive Database Monitoring ... 10-1
About Alerts ... 10-1
Performance Self-Diagnostics: Automatic Database Diagnostic Monitor ... 10-2
Monitoring General Database State and Workload ... 10-3
Managing Alerts ... 10-7
Viewing Metrics and Thresholds ... 10-7
Setting Metric Thresholds ... 10-8
About Responding to Alerts ... 10-10
Clearing Alerts ... 10-10
Setting Up Direct Alert Notification ... 10-11
Diagnosing Performance Problems Using ADDM ... 10-12
Viewing a Summary of ADDM Performance Findings ... 10-12
Responding to ADDM Performance Findings ... 10-13
Running ADDM Manually ... 10-14
Modifying AWR Snapshot Frequency and Retention ... 10-16
Using Advisors to Optimize Database Performance ... 10-16
About Advisors ... 10-16
About the SQL Advisors ... 10-18
About the Automatic SQL Tuning Advisor ... 10-19
Configuring the Automatic SQL Tuning Advisor ... 10-20
Viewing Automatic SQL Tuning Results ... 10-23
Running the SQL Tuning Advisor ... 10-24
Running the SQL Access Advisor ... 10-26
Using the Memory Advisors ... 10-27
Obtaining Memory Sizing Advice ... 10-27
Monitoring and Tuning: Oracle By Example Series ... 10-28

11 Investigating, Reporting, and Resolving Problems

About Database Problems and Incidents ... 11-1
Investigating, Reporting, and Resolving a Problem ... 11-2
Roadmap: Investigating, Reporting, and Resolving a Problem ... 11-2
Task 1 – View Critical Error Alerts in Enterprise Manager ... 11-4
Task 2 – View Problem Details ... 11-5
Task 3 – (Optional) Create a Service Request ... 11-6
Task 4 – Package and Upload Diagnostic Data to Oracle Support Services ... 11-7
Task 5 – Track the Service Request and Implement Any Repairs ... 11-8
Task 6 – Closing Incidents ... 11-9
Viewing Problems Using the Enterprise Manager Support Workbench ... 11-9

12 Managing Oracle Database Software

About Software Management and Patch Releases 12-1
Patching the Oracle Software 12-2
Determining Your Oracle Database Software Environment 12-2
Determining Your Oracle Database Version 12-2
Determining the Location of Your Oracle Home 12-2
Determining Your Hardware Configuration 12-3
Setting Your Oracle*MetaLink* Credentials 12-4
Staging and Applying Patches and Patch Sets 12-5
Viewing Critical Patch Advisories 12-8
Upgrading a Database 12-9
Overview of Database Upgrade Assistant 12-9
Pre-Upgrade Checks 12-10
Automated Upgrade Tasks 12-10
Support for Oracle Real Application Clusters 12-10
Support for Automatic Storage Management 12-10
Support for Silent Mode 12-10
Database Versions Supported by DBUA 12-10
Starting DBUA 12-11
Upgrading a Database Using DBUA 12-12
Managing Oracle Software: Oracle By Example Series 12-16

A Administering Automatic Storage Management

About Automatic Storage Management A-1
Overview of Disks, Disk Groups, and Failure Groups A-3
Installing Automatic Storage Management A-4
Accessing the Automatic Storage Management Home Page A-4
Accessing the ASM Home Page on a Single-Instance System A-4
Accessing the ASM Home Page on an Oracle RAC System A-5
Monitoring Disk Group Space Usage A-5
Creating a Disk Group A-6
Dropping a Disk Group A-8
Adding Disks to a Disk Group A-8
Dropping Disks from a Disk Group A-10
Backing Up ASM-Managed Files A-11
Automatic Storage Management: Oracle By Example Series A-12

Index

List of Figures

2–1 Oracle Universal Installer Select Installation Method Window 2-6
6–1 Oracle Database Storage Structures 6-2
6–2 Storage Options 6-2
6–3 Online Redo Log Groups and Their Members 6-4
6–4 Tablespaces Page 6-13
6–5 Home Page Space Summary Section 6-21
6–6 Segment Advisor Recommendations Page 6-21
6–7 Segment Advisor Recommendation Details Page 6-22
10–1 Database Home Page 10-3
10–2 Alerts Section of Database Home Page 10-8
10–3 SGA Size Advice 10-28
11–1 Workflow for Investigating, Reporting, and Resolving a Problem 11-3
12–1 Oracle Home Page 12-3

Preface

Oracle Database 2 Day DBA is a database administration quick start guide that teaches you how to perform day-to-day database administrative tasks. The goal of this book is to help you understand the concepts behind Oracle Database. It teaches you how to perform all common administration tasks needed to keep the database operational, including how to perform basic troubleshooting and performance monitoring activities.

Audience

Oracle Database 2 Day DBA is for anyone who wants to perform basic administrative tasks with Oracle Database. Only minimal basic knowledge of or experience with database management is required, as well as a basic knowledge of computers.

In particular, this guide is for the following groups of Oracle users:

- Developers who want to acquire basic database administrator (DBA) skills
- Anyone managing database servers
- Database administrators managing an Oracle database for a small or medium-sized business

This book is equally useful for enterprise DBAs. It recommends best practices and describes efficient ways of performing administrative tasks with Oracle Enterprise Manager as the primary interface.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an

otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Database Administrator's Guide*
- *Oracle Enterprise Manager Advanced Configuration*
- *Oracle Database Concepts*
- *Oracle Enterprise Manager Concepts*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database SQL Language Reference*
- *Oracle Database Reference*
- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* or other operating system
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Database Storage Administrator's Guide*
- *Oracle Database Installation Guide*

Many of the examples in this guide use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. See *Oracle Database Sample Schemas* for information about how these schemas were created and how you can use them.

Printed documentation is available for sale in the Oracle Store at
`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, updated Oracle documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/membership/`

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/documentation/`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

As a database administrator (DBA) for Oracle Database, you are responsible for the overall operation of Oracle Database.

This chapter contains the following sections:

- About This Guide
- About Oracle Database
- Common Oracle DBA Tasks
- Tools for Administering the Database

About This Guide

Oracle Database 2 Day DBA is a database administration quick start guide that teaches you how to perform day-to-day database administrative tasks. The goal of this guide is to help you understand the concepts behind Oracle Database. It teaches you how to perform all common administrative tasks needed to keep the database operational, including how to perform basic troubleshooting and performance monitoring activities.

The primary administrative interface used in this guide is Oracle Enterprise Manager Database Control, featuring all the self-management capabilities introduced in Oracle Database.

What This Guide Is Not

Oracle Database 2 Day DBA is task-oriented. The objective is to describe why and when administrative tasks need to be performed. Where appropriate, it describes the concepts necessary to understand and complete a task, assuming the reader has no prior knowledge of the database. This guide is not an exhaustive discussion of all Oracle Database concepts. For this type of information, see *Oracle Database Concepts*.

Additionally, for a complete discussion of administrative tasks, see *Oracle Database Administrator's Guide*.

How to Use This Guide with Related Material

This guide is part of a comprehensive set of learning material for administering Oracle Database, which includes a 2 Day DBA Oracle By Example (OBE) series, available on the Web, and in an Oracle University instructor-led class.

Every chapter in *Oracle Database 2 Day DBA* has an associated Oracle By Example lesson. The OBE steps through all tasks in the chapter and includes annotated

screenshots. In some cases, the OBE provides additional information to help you complete the task.

At the end of each chapter, you can find the link to its associated OBE lesson. The home page for the 2 Day DBA Oracle By Example series is as follows:

`http://www.oracle.com/technology/obe/11gr1_2day_dba/indexotn.html`

In addition, Oracle University provides a related instructor-led course called Oracle Database 11g: 2 day DBA. Contact Oracle University for more information.

About Oracle Database

Oracle Database is a relational database with object and Extensible Markup Language (XML) capabilities. In a relational database, all data is stored in two-dimensional tables that are composed of rows and columns. Oracle Database enables you to store data, update it, and efficiently retrieve it, with a high degree of performance, reliability, and scalability.

Oracle Database is composed of the following elements:

- The Oracle software that you install on your host computer
- The *database*, which is a collection of physical files on one or more disks

  The database contains user data, metadata, and control structures. **Metadata**, or data about the data, is the collection of information about a disk that permits Oracle software to manage user data. An example of metadata is the data dictionary. Control structures (such as the control file and redo log) ensure the integrity, availability, and recoverability of user data.
- The *Oracle instance*, which is composed of the following:
  - The background processes, which are the operating system processes or threads that perform the work of accessing, storing, monitoring, and recovering user data, metadata, and control files associated with the database
  - The shared memory areas used by the background processes
- *Server processes* that perform work on behalf of connected users and applications, and the memory and temporary storage used by these processes

  Server processes parse and execute SQL statements, and retrieve and return results to the user or application.
- Oracle Net, which is a software layer that enables client applications and Oracle Database to communicate over a network, and the Oracle Net *listener*, which is a process that listens for connection requests from the network.

See Also:

- Chapter 4, "Configuring the Network Environment"
- Chapter 5, "Managing the Oracle Instance"
- Chapter 6, "Managing Database Storage Structures"

Common Oracle DBA Tasks

As an Oracle DBA, you can expect to be involved in the following tasks:

- Installing Oracle software

- Creating Oracle databases
- Performing upgrades of the database and software to new release levels
- Starting up and shutting down the database
- Managing the storage structures of the database
- Managing users and security
- Managing database objects, such as tables, indexes, and views
- Backing up the database and performing recovery operations when necessary
- Monitoring the state of the database and taking preventive or corrective action as required
- Monitoring and tuning database performance
- Diagnosing and reporting critical errors to Oracle Support Services

In a small to medium-sized database environment, you might be the sole person performing these tasks. In large, enterprise environments, the job is often divided among several DBAs, each of whom has a specialty, such as database security or database tuning.

Tools for Administering the Database

The goal of this guide is to enable you to quickly and efficiently create an Oracle database, and to provide guidance in basic database administration.

The following are some of the products, tools, and utilities you can use to achieve your goals as a database administrator:

- **Oracle Universal Installer**

  Oracle Universal Installer (OUI) is a utility that installs your Oracle software and options. It can automatically start Oracle Database Configuration Assistant to install a database.

- **Oracle Database Configuration Assistant**

  Oracle Database Configuration Assistant (DBCA) is a utility that creates a database from templates that are supplied by Oracle, or you can create your own. It enables you to copy a preconfigured seed database, thus saving the time and effort of generating and customizing a new database.

- **Database Upgrade Assistant**

  The Database Upgrade Assistant is a tool that guides you through the upgrade of your existing database to a new Oracle Database release.

- **Net Configuration Assistant**

  Net Configuration Assistant (NETCA) is a utility that enables you to configure listeners and naming methods, which are critical components of the Oracle Database network.

- **Oracle Enterprise Manager Database Control**

  The primary product for managing your database is Oracle Enterprise Manager Database Control (Database Control), a Web-based interface. After you have installed the Oracle Database software, created or upgraded a database, and configured the network, you can use Database Control to manage your database.

Database Control also provides an interface for performance advisors and for Oracle utilities such as SQL*Loader and Recovery Manager (RMAN).

2

Installing Oracle Database and Creating a Database

This chapter describes how to install Oracle Database software and create a database.

If you are using an earlier release of Oracle Database and want to install a later release of the Oracle Database software, you can *upgrade* your existing Oracle Database and use it with the new release of the database software. See "Upgrading a Database" on page 12-9.

This chapter contains the following sections:

- Overview of Installing Oracle Database Software and Creating a Database
- Installing Oracle Database Software
- Using DBCA to Create and Configure a Database
- Installation: Oracle By Example Series

Overview of Installing Oracle Database Software and Creating a Database

To install your Oracle Database software, use Oracle Universal Installer (OUI). OUI is a graphical user interface utility that enables you to view the Oracle Database software that is already installed on your computer, install new Oracle Database software, and deinstall Oracle Database software that you no longer intend to use. Online Help is available to guide you through the installation process.

During the installation process, you are given the opportunity to create a database. If you choose to do so, OUI automatically starts Oracle Database Configuration Assistant (DBCA) to guide you through the process of creating and configuring a database.

If you do not create a database during installation, then you must run DBCA at some point after installation to create a database.

Note: After you create a database, either during installation or as a standalone operation, you do not need to create another. Each Oracle instance works with a single database only. Rather than requiring that you to create multiple databases to accommodate different applications, Oracle Database uses a single database, and accommodates multiple applications by enabling you to separate data into different *schemas* within the single database. See "About User Accounts" on page 7-1 for more information about schemas.

Checking Prerequisites

Before installing the software, OUI performs a number of automated checks to ensure that your computer fulfills the basic hardware and software requirements for an Oracle Database installation. If your computer does not meet a requirement, an error message is displayed. The requirements may vary depending upon the type of computer and operating system you are running, but some prerequisites include:

- There is a minimum of 512 MB (Microsoft Windows or Linux) or 1 GB (UNIX and other operating systems) of physical memory.
- Sufficient paging space is available.
- The appropriate service packs or patches for your operating system are installed.
- An appropriate file system format is being used.

OUI automatically sets any operating system environment variables that Oracle Database requires for its operation.

See Also:

- *Oracle Database Installation Guide* for your platform for more information about exact requirements

Installation Choices

Oracle Universal Installer guides you through an interview phase where you specify your choices for installation and database creation. The exact sequence of steps depends on your operating system.

The installation options are divided into basic and advanced. During a basic installation, you need to make only basic choices. For all other cases, you make both basic and advanced choices.

Basic Installation

When you install Oracle Database during basic and advanced installations, you need answers for the questions listed in this section. OUI provides default values for every choice.

- What database product do you want to install?

  You can install one of the following:

  - **Enterprise Edition**—This installation type is the full-featured Oracle Database product that provides data management for enterprise-level applications. It is intended for mission-critical, high-security online transaction processing (OLTP) and data warehousing environments.
  - **Standard Edition**—This installation type is suitable for workgroup or department-level applications, and for small to medium-sized enterprises. It provides core relational database management services and options.
  - **Personal Edition** (Microsoft Windows operating systems only)—This installation type installs the same software as the Enterprise Edition, but supports only a single-user, development and deployment environment.
  - **Custom Install**—This installation type enables you to customize the Enterprise Edition. Use Custom Install to add a component to your existing installation that would not usually be installed, or to prevent certain components from being installed.

- Do you want to create a database in addition to installing the Oracle Database software?

  To create the database, OUI automatically starts DBCA at the end of the installation.

  To install a preconfigured database, select one of the following database types, optimized according to usage:

  – General Purpose

  – Transaction Processing

  – Data Warehouse

  To create a custom database in which you configure your own database structure, see "Advanced Installation". Refer to Table 2–1, " DBCA Template Types" on page 2-16 for more information about the different types of preconfigured databases.

  Note: If you need to create a new database, Oracle recommends that you install a preconfigured database, which is faster and easier. You can customize the database after it has been created.

- What are your database configuration options?

  These options include your global database name and system identifier (SID), whether or not you want the Sample Schemas installed, and the passwords for the administrative accounts. The **SID** is a unique identifier that is used to distinguish this instance from other Oracle Database instances that you may create later and run concurrently on your system.

  The global database name is the full name of the database that uniquely distinguishes it from any other database. The global database name is in the form `database_name.database_domain`, for example `sales.us.acme.com`. The database name portion `sales` is a simple name you call your database. The database domain portion `us.acme.com` specifies the database domain in which the database is located. Together, the database name and domain make up the global database name.

  During basic installation, Oracle Database automatically installs the sample schemas.

  You will also be prompted to enter your database schema password for the `SYS` and `SYSTEM` accounts, which enable you to manage and administer the database.

Advanced Installation

During advanced installations, you are prompted to make the choices listed in this section, in addition to the basic options. The installation process provides default values for every choice.

The advanced installation choices are described, but the advanced installation steps are not documented in this chapter. For more information, see *Oracle Database Installation Guide* for your platform.

During an advanced installation, you are asked the following questions:

- Install the sample schemas?

To complete the exercises in this guide and related course material, you must install the sample schemas. This data is also used in most examples throughout Oracle Database documentation. Oracle recommends that you install the sample schemas.

This choice is a configuration option only during advanced installation. The default is No for this option with the advanced installation. However, sample schemas are installed by default during basic installation.

- What kind of storage mechanism do you want your database to use?

  A database includes several files that store the user data, database metadata, and information required to recover from failures. As an administrator, you decide what kind of storage subsystem to use for these files. You can select from the following options:

  – **File System**—This default option creates database files that are managed by the file system of your operating system. You can specify the directory path where database files are to be stored. Oracle Database can create and manage the actual files.

    If you are not certain about which option to use, select File System (the default).

  – **Automatic Storage Management**—This option enables you to define a pool of storage (called a disk group) in which Oracle Database automatically manages database file placement and naming. For environments with a large number of disks, this option simplifies database administration and maximizes performance. Automatic Storage Management (ASM) **stripes** the files (spreads out each database file over all disks in the disk group), optimizing disk utilization. ASM also **mirrors** (stores a separate copy of the file on a different disk in the disk group, in order to protect against loss of data in the event of disk failure) at the file level, as opposed to the disk level, for maximum storage flexibility, performance, and availability. You get the benefits of sophisticated volume managers and disk arrays without the need for expensive hardware.

    Automatic Storage Management uses an ASM instance, which is distinct from the database instance, to configure and manage disk groups. OUI guides you through the creation and starting of an ASM instance and its components. For more information, see *Oracle Database Storage Administrator's Guide*.

  – **Raw Devices**—This option enables you to manage your storage devices outside of the operating system file system by providing Oracle Database with unformatted physical disk space called raw devices. This method is primarily used in Oracle Real Application Clusters (Oracle RAC) environments. If you select this option, ensure that the raw devices were created with an operating system command, and that they are not being used for any other purpose or by other Oracle databases.

- What are your database management options?

  You specify whether to manage your database centrally or locally using Oracle Enterprise Manager. Central management enables you to manage multiple targets, such as databases and application servers, using a single interface. Using local management you can manage only a single database instance at a time.

  If you want to use central management, it requires an Oracle Enterprise Management agent to exist on each *host*, or computer that has Oracle Database software installed on it. These agents are responsible for monitoring all

components on that host. If an agent is not found on the local host, then this option is disabled during installation.

If you are setting up a single database for the first time, it is recommended that you configure local management with Oracle Enterprise Manager, which is the default. You can later install additional databases and configure central management using Enterprise Manager.

At the end of the interview, a summary window displays the options that you chose, the requirements for space based on those options, and the components that will be installed. If you chose to install a database, then this step is performed immediately after the software installation is complete.

Installing Oracle Database Software

This section describes the basic installation steps. Most steps are common to all platforms and involve running the Oracle Universal Installer. Platform-specific steps are noted.

The following steps provide a summary of the installation process. For further assistance, consult the online Help or *Oracle Database Installation Guide* for your platform.

To perform a basic installation:

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle Database software and to create and run the database.

   Refer to your operating system-specific documentation or contact your system administrator to determine whether or not you have the necessary privileges and permissions to install new software.

2. Insert the distribution media for the database into your computer.

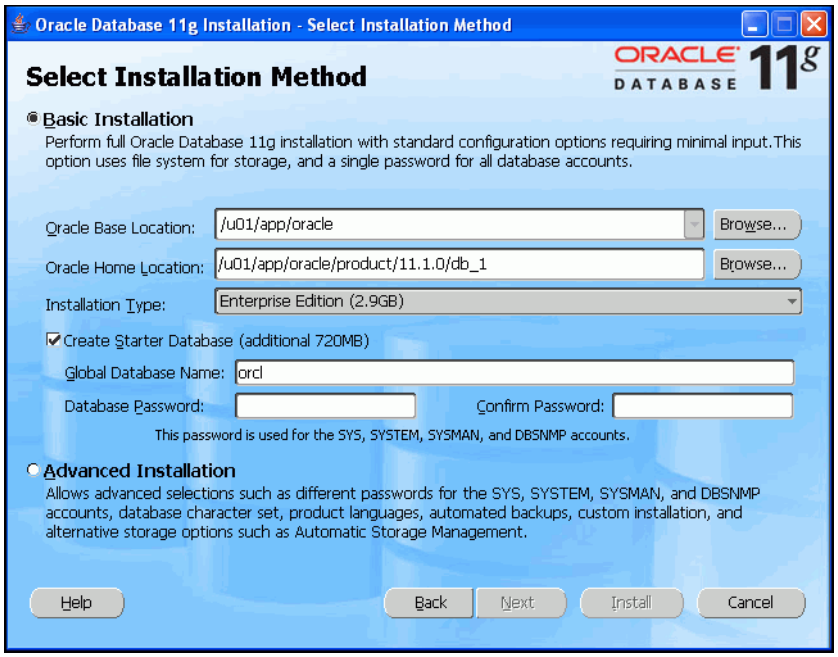
   The Autorun feature opens the Select a Product to Install window automatically.

   If you downloaded the installation software from the Oracle Web site, then follow the instructions on the site.

3. Select **Oracle Database 11g**, then click **Next**.

4. If there is no previous Oracle software installed on the computer, then the Select Installation Method window appears first, as shown in Figure 2–1.

Figure 2–1 Oracle Universal Installer Select Installation Method Window



5. Select **Basic Installation** and provide the configuration details for the database.

   In the Select Installation Method window you choose either Basic or Advanced Installation. Select the Basic Installation method to quickly install Oracle Database. This method requires minimal user input. It installs the software and optionally creates a general-purpose database based on the information you provide.

   When you choose Basic Installation you must also provide the following information:

   - **Oracle Home and Oracle Base Location**—You must specify a new Oracle home directory for each new installation of Oracle Database software.

     By default, the Oracle home directory is a subdirectory of the Oracle base directory. See the *Oracle Database Installation Guide* for your platform for more information about `ORACLE_BASE`.

     If you did not set the `ORACLE_BASE` environment variable before starting OUI, the Oracle home directory is created in an `app/`*`username`*`/` directory on the first existing and writable directory from `/u01` through `/u09` for UNIX and Linux systems, or on the disk drive with the most available space for Windows systems. If `/u01` through `/u09` does not exist on the UNIX or Linux system, then the default location is *`user_home_directory`*`/app/`*`username`*.

     You can click **Browse** to find the directory where you want to install the Oracle Database software.

   - **Installation Type**—Select either **Enterprise Edition**, **Standard Edition**, **Standard Edition One**, or **Personal Edition** (Microsoft Windows platforms only). See "Basic Installation" on page 2-2.

   - **UNIX DBA Group** (Linux and UNIX platforms only)—Specify the operating system DBA group. Refer to *Oracle Database Installation Guide for Linux*, or for your UNIX platform, for more details.

- **Create Starter Database**—Select this option to create a preconfigured database during installation. Oracle recommends that you select this option for first-time installations. You must enter a **Global Database Name** and **Database Password** for the `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP` administrator accounts. The global database name is described in "Basic Installation" on page 2-2.

You can choose the Advanced Installation option if you want to customize your installation. For example, you use this method to configure Automatic Storage Management for your database, install the Sample Schemas, configure a specific character set for your database, or configure automated backups. Selecting this option guides you through a series of installation steps that are not documented in this guide. For more information about the advanced choices, see "Advanced Installation" on page 2-3. Also see *Oracle Database Installation Guide* for your platform.

After you select Basic Installation and enter the required information, click **Next**. The Preparing to Install progress bar appears.

6. *For first time installations on Linux and UNIX operating systems only*, specify a directory for installation files and the name of an operating system group that has write permissions for that directory.

   If this is the first time you are installing any Oracle software on this computer, then the Specify Inventory Directory window appears. You must specify a local for the inventory, which OUI uses to keep track of all Oracle software installed on the computer. This information is used while applying patches or upgrading an existing installation, and while deinstalling Oracle software. Note that this directory is different from the Oracle home directory, which you specify later. The recommended value for the inventory directory is `Oracle_base/oraInventory`, where `Oracle_base` is the location of your Oracle base directory.

   In this window you can also specify the operating system group that has write permissions on the inventory directory. This prevents other users from writing over the Oracle product installation files.

   After you enter a directory path and optionally specify an operating system group, click **Next** to continue.

   The Product-Specific Prerequisite Checks window appears.

7. *For Linux and UNIX operating systems only*, after verifying that your system meets the minimum requirements for installing and configuring the chosen products, click **Next**.

   OUI performs a number of environment checks and indicates whether the check was a success, or resulted in a warning or failure. Details of the checks are provided in the displayed window. The installation can proceed only when all checks have a status of either Succeeded or Warning. If any of the environment checks failed, they must be resolved manually. See "Checking Prerequisites" on page 2-2 for more information.

   After you click Next, the Oracle Configuration Manager Registration window appears.

8. Click Next without enabling Oracle Configuration Manager.

   Oracle Configuration Manager is used to collect and upload the configuration information to the Oracle configuration repository. Using Oracle Configuration Manager can help to reduce time for resolution of support issues and allow you to

receive problem avoidance information. You can configure this option later, after the database installation is complete.

After you click Next, a summary window appears, showing information such as your global settings, space requirements, and the new products to be installed.

9. Review the installation summary, then click **Install** to start the installation.

   The Install window appears, showing the installation progress. At the end of the installation phase, the Configuration Assistants window appears. This window lists the configuration assistants that are started automatically. If you chose to create a starter database, then Database Configuration Assistant starts automatically in a separate window.

   At the end of database creation, a window is displayed that summarizes the database creation.

10. (Optional) Click **Password Management** to unlock user accounts to make the accounts accessible to users.

    The `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP` accounts are unlocked by default.

11. Click **OK** to continue the installation.

12. *For Linux and UNIX operating systems only*, run the specified scripts, then click **OK**.

    On the Execute Configuration Scripts window, you are prompted to open a new terminal window, and to run scripts as the `root` user. After you run the scripts, return to this window and click **OK**.

13. Make note of the information in the End of Installation window, then click **Exit** to exit OUI. In the confirmation window, click **Yes**.

    Your installation and database creation is now complete. The End of Installation window displays one or more important URLs, including one for the Oracle Enterprise Manager Database Control console. (The console is the user interface—the Web application—portion of Oracle Enterprise Manager Database Control.) You can go to this URL with your browser and log in as the `SYS` user with the associated password, and connect as `SYSDBA`. You use Oracle Enterprise Manager Database Control to perform common database administration tasks. See Chapter 3, "Getting Started with Database Administration".

    The URL and port information for Database Control can be found after installation in the *`Oracle_home`*`/install/portlist.ini` file.

Using DBCA to Create and Configure a Database

During installation, you can run Database Configuration Assistant (DBCA) to create and configure a database. If you already did this, you do not need to run DBCA again.

When using Oracle Database, you typically have a single database that hosts more than one application. The collection of objects that support each application are stored in different schemas and tablespaces. Information about the database objects and users for the entire database are stored in the `SYSTEM` tablespace. See Chapter 6, "Managing Database Storage Structures" for more information about tablespaces.

If you installed only the Oracle Database software, or if you want to create additional databases with the release software that you just installed, then you can use.

DBCA enables you to perform the following tasks:

- Creating a Database Using DBCA

- Changing the Configuration of a Database Using DBCA
- Deleting a Database Using DBCA
- Managing Templates with DBCA
- Configuring Automatic Storage Management Using DBCA

Online Help is available by clicking **Help**. It provides information that guides you in selecting configuration options.

Starting DBCA

You can use DBCA to create a database from templates supplied by Oracle or from templates you create.

To start DBCA:

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle Database software, and create and run the database.
2. To start DBCA on a Microsoft Windows operating system, click **Start** and then select **Programs**, **Oracle -** *home_name*, **Configuration and Migration Tools**, and then **Database Configuration Assistant**.

   To start DBCA on UNIX, or at the command-line prompt on the Windows operating system, enter the following command:

   ```
   dbca
   ```

   The `dbca` utility is typically located in the *`Oracle_home`*`/bin` directory.

   The Welcome window appears.
3. Click **Next**.

   The DBCA Operations window appears.

Creating a Database Using DBCA

On the DBCA Operations window (Step 1 of 14), select **Create a Database** to start a wizard that enables you to create and configure a database in 14 steps. The wizard requests your input in the following windows:

- Step 2 - Database Templates
- Step 3 - Database Identification
- Step 4 - Management Options
- Step 5 - Database Credentials
- Step 6 - Storage Options
- Step 7 - Database File Locations
- Step 8 - Recovery Configuration
- Step 9 - Database Content
- Step 10 - Initialization Parameters
- Step 11 - Security Settings
- Step 12 - Automatic Maintenance Tasks
- Step 13 - Database Storage

- Step 14 - Creation Options

Note that most windows of the wizard provide a default setting. Click **OK** to start the database creation.

Step 2 - Database Templates

This window enables you to select the type of database you want to create. By default, Oracle Database ships predefined templates. There are templates for Data Warehouse, General Purpose, and Transaction Processing databases.

The templates contain settings optimized for that particular type of workload. Click **Show Details** to see the configuration for each type of database. Select the template suited to the type of workload your database will support. If you are not sure which to choose, select the default **General Purpose** template.

For more complex environments, you can select the **Custom Database** option. This option results in a more extensive interview, which means that it will take longer to create your database because a database creation script must be run.

For more information about using database templates, see "Managing Templates with DBCA" on page 2-15.

Step 3 - Database Identification

In the **Global Database Name** field, enter the database name in the form *database_name.domain_name*.

In the **SID** field, enter the system identifier. The SID defaults to the database name and uniquely identifies the instance that runs the database.

Step 4 - Management Options

Use this window to set up your database so it can be managed with Oracle Enterprise Manager. Oracle Enterprise Manager provides Web-based management tools for individual databases, and central management tools for managing your entire Oracle environment.

Select **Configure Enterprise Manager** to use Enterprise Manager. Then, select one of the following options:

- If the Oracle Management Agent is installed on your host computer, then you can choose central management by selecting **Register with Grid Control for centralized management** and then selecting the Management Service.
- To manage your database locally, select **Configure Database Control for local management**. You can then optionally configure email notification of database alerts or a daily backup of the database.

Step 5 - Database Credentials

In this window, specify the passwords for the administrative accounts such as `SYS` and `SYSTEM`.

Step 6 - Storage Options

In this window, specify the type of storage mechanism you would like your database to use. For more information, see "Installation Choices" on page 2-2.

Step 7 - Database File Locations

In this window, specify the Oracle software home and the directory path in which to create the database files. Select one of the following options:

- **Use Database File Locations from Template**—This option instructs DBCA to use the directory information as specified in the template. Later, you can make modifications to database file names and locations.
- **Use Common Location for All Database Files**—This option requires you to specify a new directory for the Oracle home. All the database files will be created in this location. Later, you can make modifications to database file names and locations.
- **Use Oracle-Managed Files**—This option instructs Oracle Database to directly manage operating system files comprising an Oracle database. You specify the default location, called a database area, for all your files. Oracle Database thereafter automatically creates and deletes files in this location, as required. You can also create multiple copies of your redo and online log files by selecting **Multiplex Redo Logs and Control Files**. To learn more about redo logs and control files, see Chapter 9, "Performing Backup and Recovery".

  Selecting this option enables you to delegate the complete management of database files to the database. You no longer need to specify the file names, location, or their sizes.

Step 8 - Recovery Configuration

When you create a new database, it is important to configure the database so you can recover your data in the event of a system failure. Redo logs contain a record of changes that were made to datafiles. Redo logs are stored in redo log groups, and you must have at least two redo log groups for your database. After the redo log files in a group have filled up, the log writer process (LGWR) switches the writing of redo records to a new redo log group. Oracle Database can automatically save the inactive group of redo log files to one or more offline destinations, known collectively as the **archived redo log** (also called the archive log). The process of turning redo log files into archived redo log files is called **archiving**.

Archiving can be performed only if the database is running in `ARCHIVELOG` mode. This means a group of redo log files cannot be reused by the log writer (LGWR) process until the group is archived. If the database is running in `NOARCHIVELOG` mode, then when a group becomes inactive after the LGWR process switches to a new group, the inactive group is available for immediate reuse by the LGWR process.

The `NOARCHIVELOG` mode protects a database from instance failure but not from media failure. Only the most recent changes made to the database, which are stored in the online redo log groups, are available for instance recovery. To restore a database operating in `NOARCHIVELOG` mode, you can use only entire database backups taken while the database is closed. Therefore, if you decide to operate a database in `NOARCHIVELOG` mode, back up the entire database at regular, frequent intervals.

The archiving of redo log files has these advantages:

- A database backup, together with online and archived redo log files, guarantees that you can recover all committed transactions in the event of an operating system or disk failure.
- You can recover the database using a backup that was taken while the database was open and being used, as long as you have a copy of the archived log files that were written while the database was being backed up.

- You can perform online tablespace backups, and use these backups to restore a tablespace following media failure.
- You can keep a standby database current with its original database by continuously applying the original archived redo logs to the standby database.

Before you can archive redo logs, you must determine the destination to which you will archive. Oracle recommends that the archive log be stored in a flash recovery area because it can simplify backup and recovery operations for your database. A **flash recovery area** is a location in which Oracle Database can store and manage files related to backup and recovery. It is distinct from the database area, which is a location for the current database files (datafiles, control files, and online redo logs).

When creating your database, you can select the following options:

- **Specify Flash Recovery Area**—Select this option to specify a backup and recovery area and its directory location and size. You can use variables to identify standard locations.
- **Enable Archiving**—Select this option to enable the archiving of database redo logs, which can be used to recover a database. Selecting this option is the same as enabling Archive Log Mode in Oracle Enterprise Manager Database Control or running the database in `ARCHIVELOG` mode.

  Oracle recommends you select **Enable Archiving**. Selecting this option provides better protection for your database in the case of software or hardware failure. If you do not select this option now, you can enable ARCHIVELOG mode later. See "Configuring Your Database for Basic Backup and Recovery" on page 9-4.

Step 9 - Database Content

When you create your database, you can load it with data using either of the following methods:

- **Sample Schemas—**Select **Sample Schemas** to include the Sample Schemas (`EXAMPLE`) tablespace in your database. Oracle guides and educational materials contain examples based upon the Sample Schemas. Oracle recommends that you include them in your database.
- **Custom Scripts—**In the **Custom Scripts** subpage, optionally specify one or more SQL scripts to run after your database is created. Scripts are useful for performing postinstallation tasks, such as loading custom schemas. Note that if you choose to run scripts after installation, your scripts must include a connection string that identifies the database. Click **Help** for more information.

Step 10 - Initialization Parameters

The links in this window provide access to additional windows that enable you to change the default initialization parameter settings. These parameters fall into the following categories:

- Memory
- Sizing
- Character Sets
- Connection Mode

You can also click the **All Initialization Parameters** button at the bottom of the window to display a list of all the database initialization parameters and their current settings.

Memory Use this window to set the initialization parameters that control how the database manages its memory. You can choose from one of the following methods for memory management:

- **Typical**—This method requires little configuration, and allocates memory as a percentage of total overall physical system memory. Select **Typical** and enter a percentage value. Click **Show Memory Distribution** to see how much memory DBCA assigns to the System Global Area (SGA) and the aggregate Program Global Area (PGA). With this setting, the system automatically tunes many of the memory components of the SGA, and allocates memory to individual PGAs as needed. The system can also dynamically decrease or increase the total amount of memory allocated to the SGA or aggregate PGA, depending on processing demands. The total memory used for the database instance never exceeds the amount you specify. This automatic memory tuning for the instance is known as **automatic memory management**. To learn more about PGA and SGA, see "About Instance Memory Structure" on page 5-4.
- **Custom**—This method requires more configuration than the Typical option, but gives you more control over how the database uses system memory. This option is meant for more experienced database administrators. You can directly specify memory sizes for the SGA and aggregate PGA and their substructures, such as the shared pool and buffer cache.

  Select one of the following options for customized memory management:

  – **Automatic Shared Memory Management** to allocate specific amounts of memory to the SGA and aggregate PGA. With this setting, automatic shared memory management is enabled for the SGA, and memory is allocated to the individual PGAs as needed.

  – **Manual Shared Memory Management** to enter specific values for each SGA component and the aggregate PGA. This disables automatic shared memory management and enables you to determine how the SGA memory is distributed among the SGA memory components.

  See Also:

  - Chapter 5, "Managing the Oracle Instance" for more information about memory management options

Sizing In this tab, you specify the smallest block size and the maximum number of operating system user processes that can simultaneously connect to the database.

- **Block Size**—Use this list to select the block size, or accept the default. Oracle Database data is stored in data blocks of the size specified. One data block corresponds to a specific number of bytes of physical space on disk. Selecting a block size other than the default 8 kilobytes (KB) value requires advanced knowledge and should be done only when absolutely required.

  While using predefined templates, this list is not enabled because the database will be created with the default block size of 8 KB. While using the custom option, you can change the block size.
- **Processes**—In this field, specify the maximum number of processes that can simultaneously connect to the database. Enter a number or accept the default value of 150. The default value for this parameter is appropriate for many environments. The value you select should allow for all background processes, user processes, and parallel execution processes.

Character Sets Use this tab to define the character sets used by your database. **Character sets** are the encoding schemes used to display characters on your computer screen. The character set determines what languages can be represented in the database.

Oracle recommends Unicode AL32UTF8 as the database character set. **Unicode** is the universal character set that supports most of the currently spoken languages of the world. The default character set suggested or used by Oracle Universal Installer and Database Configuration Assistant is based on the language configuration of the local operating system. If you choose to use a different character set, choose the character set most frequently used by the clients connecting to this database, for example WE8MSWIN1252. You might choose this character set if most clients connecting to the database run under the Microsoft Windows operating system.

- **Database Character Set—**In this section, select from one of the following options:
  - **Use the Default—**Select this option to support only the language currently used by the operating system for all your database users and database applications.
  - **Use Unicode (AL32UTF8)**—Select this option to support multiple languages for your database users and database applications.
  - **Choose from the list of character sets—**Select this option if you want Oracle Database to use a character set other than the default character set used by the operating system.
- **National Character Set—**In this list, select a character set or accept the default. The national character set is an alternative character set that enables you to store Unicode characters in a database that does not have a Unicode database character set. Choosing a national character set can make programming for multiple languages easier.
- **Default Language**—In this list, select a default database language or accept the default. The default language determines how the database supports locale-sensitive information such as day and month abbreviations, default sorting sequence for character data, and reading direction (left to right or right to left).
- **Default Territory—**In this list, select the name of the territory whose conventions are to be followed for day and week numbering or accept the default. The default territory also establishes the default date format, the default decimal character and group separator, and the default International Standardization Organization (ISO) and local currency symbols. For example, in the United Kingdom, the default date format is DD-MON-YYYY, where DD is the day of the month (1-31), MON is the abbreviated name of the month, and YYYY is the 4-digit year.

Connection Mode Use this window to select the database mode. You can run the database in either of the following modes:

- **Dedicated Server Mode**—This mode allows a dedicated server process for each user process. Select this option when the number of total clients is expected to be small, for example, 50 or fewer. You might also choose this option when database clients will typically make persistent, long-running requests to the database. By default, the database is configured for dedicated server processes.
- **Shared Server Mode—**This mode allows several client connections to share a database-allocated pool of resources. Use this mode in configurations in which client load is expected to cause a strain on memory and other system resources. If you choose shared server mode, then you must indicate the number of server

processes you want to create when a database instance is started. For more information about setting this parameter, click **Help**.

Step 11 - Security Settings

In this window, choose to use either the enhanced default security settings or the settings of a previous release, optionally disabling certain security features.

Step 12 - Automatic Maintenance Tasks

On this page you decide whether or not you want to enable the automatic maintenance tasks feature of Oracle Database 11*g*.

Step 13 - Database Storage

In this window, a navigation tree displays the storage structure of your database (control files, datafiles, redo log groups, and so forth). If you are not satisfied with the storage structure or parameters, then you can make changes.

Note that if you selected one of the preconfigured templates for a database, then you cannot add or remove control files, datafiles, or undo segments.

Step 14 - Creation Options

In this window, select any of the following options for creating the database:

- **Create Database**—Select this option to create your database at this time.
- **Save as a Database Template**—Select this option to save the database definition as a template to use at a later time.
- **Generate Database Creation Scripts**—Select this option to generate a SQL database creation script that you can run at a later time.

After you make your choice, click **Finish**. A confirmation window appears after a short period of time. Review the information, and click **OK** to proceed.

Changing the Configuration of a Database Using DBCA

In the Operations window, select **Configure Database Options** to change various aspects of your database configuration. For example, you can change the database connections from dedicated server to shared server. You can add database options that have not been previously configured for use with your database, for example, Oracle Label Security or Oracle OLAP.

Deleting a Database Using DBCA

In the Operations window, select **Delete a Database** to remove a database from the system. When you select this option, DBCA deletes all the files associated with this database. With the Windows operating system, any associated services are also deleted.

Managing Templates with DBCA

DBCA templates are XML files that contain information required to create a database. Templates are used in DBCA to create new databases and duplicate existing databases. The information in templates includes database options, initialization parameters, and storage attributes (for datafiles, tablespaces, control files, and online redo logs).

Templates can be used just like scripts, but they are more powerful than scripts because you have the option of duplicating a database. Duplication saves time because you copy the files of an existing database, referred to as a *seed database*, to the correct locations.

Templates are stored in the following directory:

Oracle_home/assistants/dbca/templates

Advantages of Using Templates

Using templates has the following advantages:

- Time saving. If you use a template, you do not have to define the database.
- Easy duplication. By creating a template containing your database settings, you can easily create a duplicate database without specifying parameters twice.
- Easy editing. You can quickly change database options from the template settings.
- Easy sharing. Templates can be copied from one computer to another.

Types of Templates

Templates are divided into the following types:

- Seed templates
- Nonseed templates

The characteristics of each are shown in Table 2–1.

Table 2–1 DBCA Template Types

Type	File Extension	Includes Datafiles	Database Structure
Seed	.dbc	Yes	This type of template contains both the structure and the physical datafiles of an existing database, referred to as the *seed database*. Your new database starts as a copy of the seed database, and requires only the following changes: ■ Name of the database ■ Destination of the datafiles ■ Number of control files ■ Number of redo log groups ■ Initialization parameters Other changes can be made after database creation using custom scripts that can be invoked by DBCA, command-line SQL statements, or Oracle Enterprise Manager Database Control. The datafiles and online redo logs for the seed database are stored in a compressed format in a file with a .dfj extension. The corresponding .dfj file location is stored in the .dbc file.
Nonseed	.dbt	No	This type of template is used to create a new database. It contains the characteristics of the database to be created. Nonseed templates are more flexible than their seed counterparts because all datafiles and online redo logs are created to your specification, and names, sizes, and other attributes can be changed as required.

DBCA Templates Provided by Oracle Database

Oracle Database provides templates for the environments shown in Table 2–2.

Table 2–2 DBCA Template Environments

Environment	Description of Environment
Data warehouse	Users perform numerous, complex queries that process large volumes of data. Response time, accuracy, and availability are key issues. These queries (`SELECT` statements) range from a fetch of a few records to queries that sort thousands of records from many different tables.
Transaction processing	Many concurrent users perform numerous transactions that require rapid access to data. Availability, speed, concurrency, and recoverability are key issues. Transactions consist of reading (`SELECT` statements), writing (`INSERT` and `UPDATE` statements), and deleting (`DELETE` statements) data in database tables.
General purpose	This template creates a database designed for general use. It combines features of both the Data warehouse and Transaction processing database templates.
Custom database	This template allows you maximum flexibility in defining a database because you can change any of the settings for the database being created.

Creating Templates Using DBCA

On the Operations page, select **Manage Templates** and click **Next** to display the Manage Template window. This window provides you with the option of creating or deleting a template. DBCA saves templates as XML files.

To create a database template, select one of the following options:

- **From an existing template**

  Using an existing template, you can create a new template based on the predefined template settings. You can add or change any template settings such as initialization parameters, storage parameters, or whether or not to use custom scripts.

- **From an existing database (structure only)**

  You can create a new template that contains structural information from an existing database, including database options, tablespaces, datafiles, and initialization parameters. User-defined schemas and their data *will not* be part of the created template. The source database can be either local or remote. Select this option when you want the new database to be structurally similar to the source database, but not contain the same data.

- **From an existing database (structure as well as data)**

  You can create a new template that has both the structural information and physical datafiles of an existing database. Databases created using such a template are identical to the source database. User-defined schemas and their data *will* be part of the created template. The source database must be local. Select this option when you want to create an exact replica of the source database.

When creating templates from existing databases, you can translate file paths into Optimal Flexible Architecture (OFA) or maintain existing file paths. OFA is a set of file naming and placement guidelines for Oracle software and databases. Using OFA is recommended if the target computer on which you plan to create a database using the template has a different directory structure than computer on which the template was

defined. Standard file paths can be used if the target computer has a directory structure that is similar to the directory structure on the source computer.

Deleting DBCA Templates

The Template Management window enables you to view and delete existing templates. When you delete a template, it is no longer available to create a new database or a new template. Select a template and click **Finish** to remove it from the list of current templates.

Configuring Automatic Storage Management Using DBCA

In the Operations window, select **Configure Automatic Storage Management** to configure Automatic Storage Management (ASM). For a brief overview of ASM, see "Advanced Installation" on page 2-3. For more detailed information, see *Oracle Database Storage Administrator's Guide*.

Note: Before you can use ASM, you must configure a Cluster Synchronization Services (CSS) process by running the following command as the `root` user:

```
Oracle_home/bin/localconfig add
```

Installation: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Installation OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/install/install.htm
```

3

Getting Started with Database Administration

This chapter provides a brief roadmap for administering your database, and introduces you to Oracle Enterprise Manager Database Control (Database Control), the Web-based interface for managing an Oracle database.

This chapter contains the following sections:

- Managing Your Database: A Roadmap
- Introduction to Oracle Enterprise Manager Database Control
- Starting and Stopping the Database Control Console Process
- Accessing the Database Home Page
- Access to Database Control for Nonadministrative Users
- Creating Database Control Administrative Users
- Setting Database Control Preferences
- About SQL and SQL-Based Management Tools
- Getting Started with Oracle Enterprise Manager: Oracle By Example Series

Managing Your Database: A Roadmap

This section provides an overview of the tasks involved in managing an Oracle database instance. Each task is described in detail in a different chapter in this guide.

To manage your Oracle database:

1. Start the database instance. At the end of the installation, your instance is already started and your database is open. In the future, there will be times, perhaps for doing database maintenance or because of a power or media failure, that you shut down your database instance and later restart it.

   See "Shutting Down and Starting Up the Oracle Instance" on page 5-7.

2. Optionally, configure the network environment to enable clients to connect to your database.

   See Chapter 4, "Configuring the Network Environment".

3. Review your database storage structure: tablespaces and datafiles, redo log files, and control files. Create or modify storage structures as needed.

   See Chapter 6, "Managing Database Storage Structures".

4. Review memory allocation and adjust as needed.

   See "Managing Memory" on page 5-12.

5. Review, unlock, and reset passwords for predefined database users as needed. Create new users, and assign privileges and roles to them.

   See Chapter 7, "Administering User Accounts and Security".

6. Create any necessary schema objects, including tables, views, and indexes. Populate the tables with data.

   See Chapter 8, "Managing Schema Objects".

7. Create or review the backup strategy for the database and back up the database.

   See Chapter 9, "Performing Backup and Recovery".

8. Enable archiving of redo logs, if not already done.

   See "Configuring Recovery Settings" on page 9-6.

9. Monitor database performance, diagnose performance problems, and tune the database as necessary.

   See Chapter 10, "Monitoring and Tuning the Database".

10. Investigate, gather diagnostic data for, and report to Oracle Support Services any critical database errors.

    See Chapter 11, "Investigating, Reporting, and Resolving Problems".

11. Keep Oracle Database software up-to-date with the latest patch releases.

    See Chapter 12, "Managing Oracle Database Software".

Introduction to Oracle Enterprise Manager Database Control

The Web-based Oracle Enterprise Manager Database Control (Database Control) is the primary tool for managing your Oracle database. It is installed with the database.

The following topics introduce you to Database Control:

- Database Management Features
- Database Home Page
- Online Help
- Navigational Features

Database Management Features

Using Database Control, you can perform administrative tasks such as creating schema objects (tables, views, indexes, and so on), managing user security, managing database memory and storage, backing up and recovering your database, and importing and exporting data. You can also view performance and status information about your database.

Database Home Page

The main Database Control page for database administration is called the Database Home page. This is the page that loads when you log in to Database Control.

See "Accessing the Database Home Page" on page 3-4 for more information.

Online Help

You can access context-sensitive online Help by clicking the **Help** link displayed at the top of every page. On any Help page, click **Contents** for links to all Help topics. A search facility enables you to search the contents of Help.

Navigational Features

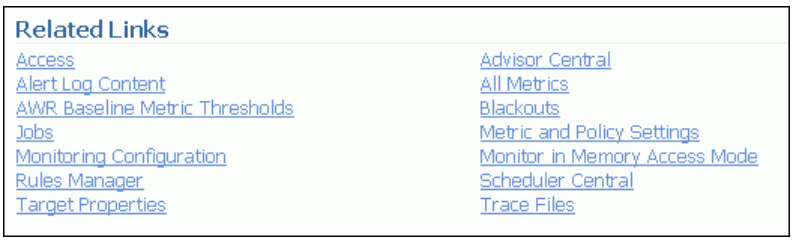
Navigational features of the Database Home page include the following:

- Subpage links at the top of the page. These links take you to the various Database Control subpages that organize management tasks into distinct categories. For example, you click **Availability** to perform tasks associated with backing up and recovering your data.

- Drill-down links that provide increasing levels of detail.

  For example, clicking the link next to the label Host provides additional detail about the computer on which the database runs.

- Suggested related links for more information.



Starting and Stopping the Database Control Console Process

To access Oracle Enterprise Manager Database Control (Database Control) from a client browser, the Database Control console process `dbconsole` needs to be running on the database host computer. The `dbconsole` process is automatically started after installation. However, in the event that the system needs to be restarted, you can start it manually at the command line or start it as a service in Windows. You can also run commands to stop the `dbconsole` process or view its status. If the `dbconsole` process is stopped, it must be manually restarted before you use the Database Control console.

To start or stop, or to view the status of the dbconsole process from the command line:

1. Open a operating system command window and go to your *`Oracle_home`*`/bin` directory.
2. Do one of the following:
   - To start the `dbconsole` process, run the following command:

     ```
     ./emctl start dbconsole
     ```

- To stop the `dbconsole` process, run the following command:

  ```
  ./emctl stop dbconsole
  ```

- To view the status of the `dbconsole` process, run the following command:

  ```
  ./emctl status dbconsole
  ```

To start or stop dbconsole as a service on Microsoft Windows:

1. Do one of the following:
   - On Windows XP and Windows Server 2003, click **Start**, and then select **Control Panel**.
   - On Windows 2000 Server, click **Start**, select **Settings**, and then select **Control Panel**.

   The Control Panel window opens.
2. Double-click the **Administrative Tools** icon, and then double-click the **Services** icon.

   The Services window opens. Oracle Database services begin with `Oracle`. The `dbconsole` service is listed as `OracleDBConsole`*`Oracle_sid`*, where *`Oracle_sid`* is the system identifier (SID) for your database instance. The status of this process (either Started or Stopped) is listed in the Status column.
3. Double-click the `OracleDBConsole`*`Oracle_sid`* service.

   The service properties window opens.
4. In the Startup Type list, ensure that either **Manual** or **Automatic** is selected.
5. Do one of the following:
   - To start `dbconsole`, click **Start**, and then click **OK**.
   - To stop `dbconsole`, click **Stop**, and then click **OK**.

Accessing the Database Home Page

The Database Home page is the main database management page in Oracle Enterprise Manager Database Control (Database Control).

To access the Database Home page:

1. Ensure that the `dbconsole` process is running on the database host computer.

   See "Starting and Stopping the Database Control Console Process" on page 3-3.
2. In your Web browser, enter the following URL:

   `https://`*`hostname:portnumber`*`/em`

   For example, if you installed the database on a host computer named comp42.mycompany.com, and the installer indicated that your Enterprise Manager Console HTTP port number is 1158, enter the following URL:

   ```
   https://comp42.mycompany.com:1158/em
   ```

   You can determine the port number for Database Control on Linux and UNIX systems by viewing the *`Oracle_home`*`/install/portlist.ini` file. On Microsoft Windows systems, you can determine the URL for Database Control by

viewing the Database Control Properties window. To view this window, use the Start menu, navigate to the Database Control entry in the Oracle home folder, then right-click this entry and select **Properties**.

When you access Database Control, if the database is running, it displays the Login page. If the database is down and needs to be restarted, Database Control displays the Startup/Shutdown and Perform Recovery page.

You must do the following to start the database:

a. Click **Startup/Shutdown**, enter the host login user name and password, and then enter the database login user name and password.

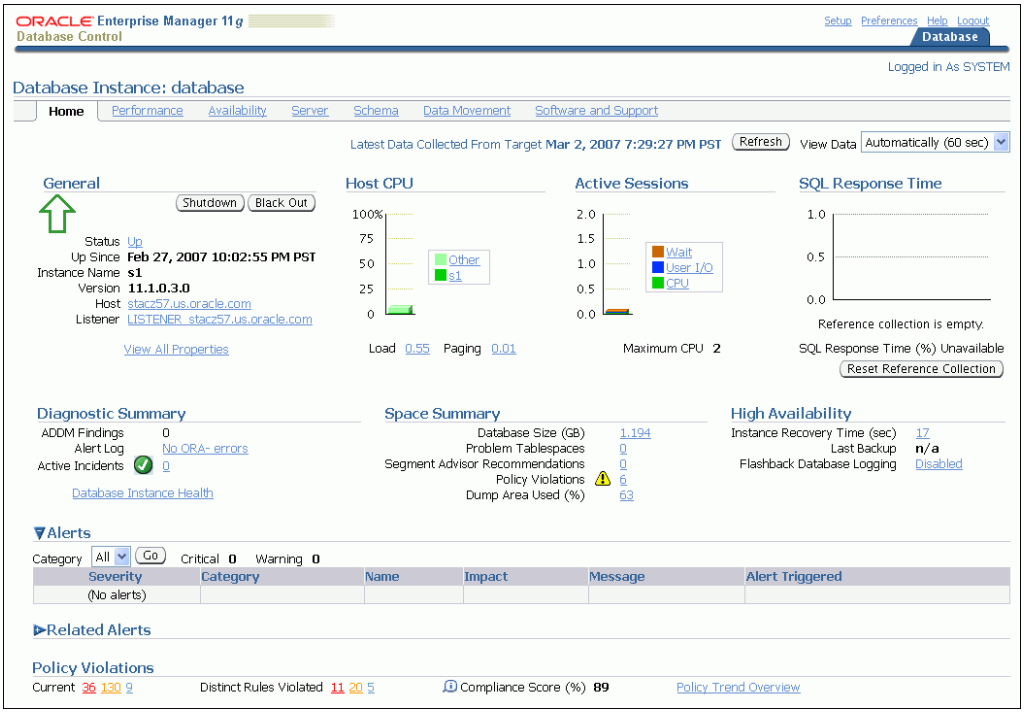
For the database user name and password, use `SYS` and the password that you specified during installation.

b. Click **OK** to start the database, and then in the Confirmation page, click **Yes** to start the database in open mode.

3. Log in to the database with a user account that is authorized to access Database Control. This initially could be `SYS` or `SYSTEM`, with the password that you specified during database installation.

For typical day-to-day administrative tasks, it is recommended that you log in with the `SYSTEM` account. If you want to back up, recover, or upgrade the database, you must log in with the `SYS` account.

Database Control displays the Database Home page.



The subpage links across the top of the page enable you to access performance, availability, and other administration pages to manage your database. The features provided by these pages are discussed in other sections of this guide.

The various sections of the Database Home page provide information about the environment and status of the database. For example, the Alerts and Diagnostic Summary sections warn you of errors and performance problems that are impacting the operation of your database. You can click the provided links to see more detail

about the problem areas, and, in some cases, to obtain recommendations for resolving the problems. This is discussed in Chapter 10, "Monitoring and Tuning the Database".

See Also:

- "Starting and Stopping the Database Control Console Process" on page 3-3
- *Oracle Database 2 Day + Security Guide*

Access to Database Control for Nonadministrative Users

As a database administrator, you can log in to Oracle Enterprise Manager Database Control (Database Control) with the `SYS` or `SYSTEM` user account to perform administrative and other tasks. Nonadministrative users may also want to log in to Database Control. For example, application developers may want to take advantage of the Database Control graphical user interface to create or modify tables, indexes, views, and so on. You must grant access to Database Control to these users before they can log in.

For nonadministrative users to have access to Database Control, they must be granted the `SELECT ANY DICTIONARY` system privilege. See "Example: Granting Privileges and Roles to a User Account" on page 7-13.

See Also: *Oracle Database 2 Day + Security Guide*

Creating Database Control Administrative Users

When you log in to Oracle Enterprise Manager Database Control (Database Control) using the `SYS`, `SYSTEM`, or `SYSMAN` user accounts, you are logging in as the Database Control super user. These are the only accounts that are automatically granted the roles and privileges required to administer Database Control itself. Examples of Database Control administration tasks include the following:

- Creating other Database Control administrators
- Configuring e-mail and other notification methods

  See "Setting Up Direct Alert Notification" on page 10-11.
- Setting alert thresholds for database metrics, such as *tablespace space usage percentage exceeded* or *SQL response time exceeded*

  See "Managing Alerts" on page 10-7.
- Selecting database policies to apply, so that Database Control can show alerts if a policy is violated. (An example policy is "A nonsystem user cannot use the `SYSTEM` or `SYSAUX` tablespace as its default tablespace.")
- Defining **blackouts**, which are time periods in which database monitoring is suspended so that maintenance operations do not skew monitoring data or generate needless alerts.

  See "Defining Blackout Periods" on page 3-8.

You can create Database Control administrative users who have enough privileges to administer Database Control itself, but lack the high-level database administration privileges of the `SYS` and `SYSTEM` users. This enables you to assign the minimum privileges required for other Database Control administrators to do their jobs, which is a best practice for database security. You can also create a Database Control

administrative account for yourself, thus avoiding logging in as SYS or SYSTEM until you must perform database administration tasks.

Using the following procedure, you can assign Database Control administrative privileges to an existing database user or create a new Database Control administrative user. When you create a new Database Control administrative user, a user account is created for that user for the database. You must then decide which system privileges, object privileges, or roles to grant the user, if any, to perform database administration tasks.

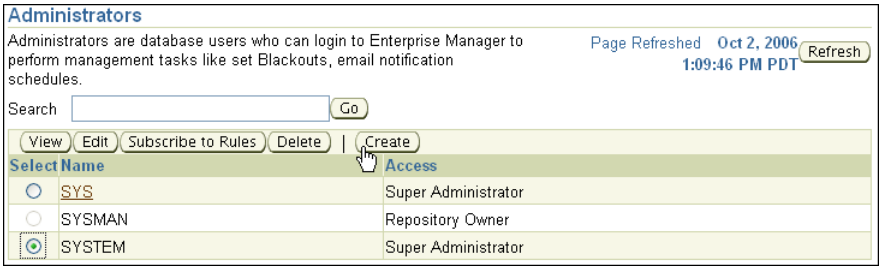
To create a Database Control administrative user:

1. On any Database Control page, at the top of the page, click **Setup**.

   The Enterprise Manager Configuration page appears, showing the Overview of Setup page.

2. In the left navigation bar, click **Administrators**.

   The Administrators page appears.



3. Click **Create**.

   The Create Administrator: Properties page appears.

4. In the **Name** field, enter a new user name, or click the flashlight icon next to the field and select an existing database user.

5. In the **Password** and **Confirm Password** fields, enter a user password.

   If you are creating a new user, assign a password. If you are selecting an existing user, enter any text in the password fields. The text is ignored, and the password for the user is not changed.

6. Enter one or more e-mail addresses for this administrator only if you plan to set up e-mail notifications for the database.

   See "Setting Up Direct Alert Notification" on page 10-11 for more information.

7. Click **Review** to view a page that summarizes the information that you entered.

8. Click **Finish** to create the new administrative user.

The Administrators page appears, showing the new administrator in the list.

Setting Database Control Preferences

This section discusses setting user preferences for Oracle Enterprise Manager Database Control (Database Control). It contains the following topics:

- About Database Control Preferences
- Defining Blackout Periods
- Setting Preferred Credentials

About Database Control Preferences

Oracle Enterprise Manager Database Control (Database Control) enables you to set user preferences in the following areas:

Notification

These settings enable Database Control to e-mail you alerts. Alerts are notifications that the database is in an undesirable state and needs your attention. By default, the Database Home page lists all alerts. However, setup is required for e-mail notification. For more information about alerts and setting up notifications, see "Setting Up Direct Alert Notification" on page 10-11.

Blackout Administration

Blackouts suspend the collection of database monitoring data and the sending of alerts by Database Control. This enables you to perform scheduled maintenance on the database without receiving needless alerts and without skewing the monitoring data. For example, you can stop data collections during a database backup or a hardware upgrade. If you continue monitoring during these periods, the collected data will show trends and other monitoring information that are not the result of typical day-to-day operations. To get a more accurate, long-term picture of database performance, you can use blackouts to exclude these special-case situations from data analysis. See "Defining Blackout Periods" on page 3-8 for more information.

Preferred Credentials

Database Control can automatically execute many routine administrative tasks, such as backups. This is done using a job scheduling system built into Database Control. To keep your environment secure, setting up tasks for automatic execution in Enterprise Manager requires you to provide login information for the host computer and database. To avoid entering this information every time you create or run a job or task, Database Control enables you to save this information as preferred credentials. Preferred credentials are stored in the database in encrypted mode to protect them from unauthorized use. See "Setting Preferred Credentials" on page 3-10.

Defining Blackout Periods

You can define one or more one-time or repeating blackout periods during which collection of database monitoring information and the sending of alerts are suspended.

To define a blackout period:

1. On any Database Control page, at the top of the page, click **Setup**.

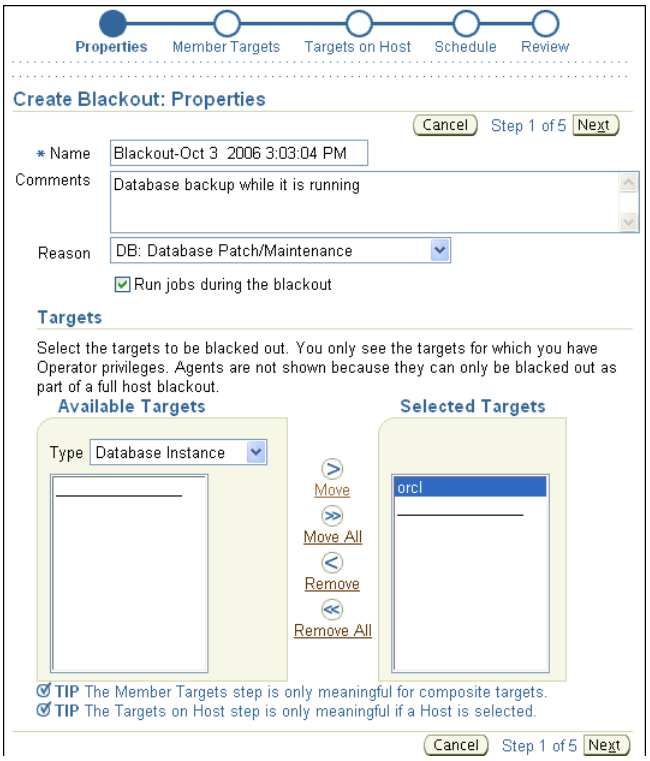
   The Enterprise Manager Configuration page appears, showing the Overview of Setup page.

2. In the left-hand pane, click **Blackouts**.

   The Blackouts page appears.

3. Click **Create** to start the Create Blackout wizard.

   The Create Blackout: Properties page appears.



4. (Optional) Replace the default blackout name with one of your choosing.

5. (Optional) In the **Comments** field, enter text that describes the purpose of the blackout.

6. In the Reason list, select the blackout reason that is most appropriate.

7. In the Available Targets section, in the Type list, select **Database Instance**.

   The system identifier (SID) for your database instance appears in the Available Targets list.

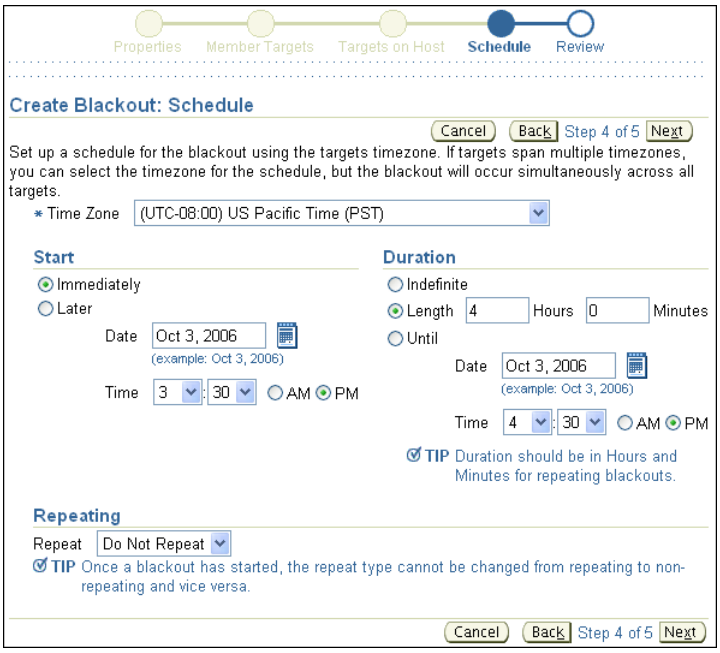
8. In the Available Targets list, select your instance SID, and then click the **Move** icon.

   Note: You can also double-click the instance SID.

   The instance SID is moved from the Available Targets list to the Selected Targets list.

9. Click **Next**.

   The Create Blackout Schedule page appears.



10. In the Start section, schedule the blackout either immediately or for a later date and time.
11. In the Duration section, indicate the duration of the blackout.
12. To repeat the blackout periodically, in the Repeating section, select a repeat frequency from the Repeat list.
13. Click **Next**.

    The review page appears.

    Review what you have entered. You can click **Back** to change a setting.
14. Click **Finish**.

    The Confirmation page appears, with the new blackout period shown in the list.

See Also:

- "About Alerts" on page 10-1

Setting Preferred Credentials

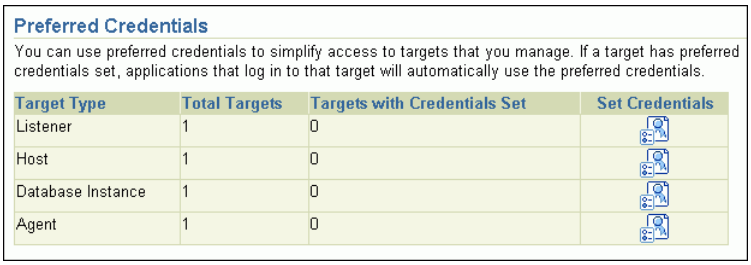
When you set preferred credentials, Database Control automatically fills in host computer and database login credentials for you at times when it usually prompts for these credentials, or when it is about to run a job that requires credentials.

To set preferred credentials for the database:

1. From any Database Control page, at the top of the page, click **Preferences**.

   The Preferences page appears.
2. Click **Preferred Credentials** in the left-hand pane.

   The Preferred Credentials page appears, showing a table of targets.


Preferred Credentials

You can use preferred credentials to simplify access to targets that you manage. If a target has preferred credentials set, applications that log in to that target will automatically use the preferred credentials.

Target Type	Total Targets	Targets with Credentials Set	Set Credentials
Listener	1	0	
Host	1	0	
Database Instance	1	0	
Agent	1	0	

3. In the table row for the Database Instance, click the icon under **Set Credentials**.

   The Preferred Credentials page appears.

4. For your database, enter the credentials for **Normal Username/Password**, **SYSDBA Username/Password**, and **Host Username/Password**. For example, you might typically connect as the user `SYSTEM`, use the `SYS` account for SYSDBA access, and use `oracle` for the host user name.

5. Click **Test** to test your credentials.

   A confirmation message is displayed if your credentials can be verified.

6. Click **Apply** to apply the changes.

About SQL and SQL-Based Management Tools

Many of the operations performed by Oracle Enterprise Manager Database Control (Database Control) use structured query language (SQL) statements. SQL (pronounced like *sequel*) is an industry-standard English-like computer programming language for querying and updating databases.

The following is an example of a SQL query that lists information about countries in a countries table, which is owned by user `hr`:

```
SELECT COUNTRY_ID, COUNTRY_NAME FROM HR.COUNTRIES;
```

SQL is a powerful language that can also be used to perform a variety of database administration tasks. The following SQL statement creates the database user `nick` and assigns him a password of your choosing, represented by *password*:

```
CREATE USER nick IDENTIFIED BY password;
```

When performing an administrative task in Database Control, you can click **Show SQL** to see the SQL statements being issued.

In addition to using the graphical management pages presented in Database Control, you can use other Oracle tools such as SQL Developer and SQL*Plus to issue SQL statements. These tools enable you to perform database management operations, and to query, insert, update, or delete data directly in the database.

Starting SQL*Plus and Connecting to the Database

SQL*Plus is a command-line program that you use to submit SQL and PL/SQL statements to an Oracle database. You can submit statements interactively or as SQL*Plus scripts. SQL*Plus is installed with the database and is located in your *Oracle_home*`/bin` directory.

You can start SQL*Plus from the command line, or on Microsoft Windows, from the Start menu.

To start SQL*Plus and connect to the database from the command line:

1. Open a command window.
2. (Linux and UNIX systems only) Set the required environment variables (`ORACLE_HOME` and `ORACLE_SID`) by running the script `oraenv` (for the Bourne, Korn, or Bash shell) or `coraenv` (for the C shell).

   These scripts are typically located in the `/usr/local/bin` directory. They may prompt you for the desired value of the `ORACLE_SID` environment variable. If so, supply the system ID (SID) that you chose when you installed Oracle Database. The default SID that the installer typically suggests is `orcl`.
3. (Linux and UNIX systems only) Either ensure that the *`Oracle_home`*`/bin` directory is in your `PATH` environment variable, or change to the *`Oracle_home`*`/bin` directory.
4. Enter the following command:

   ```
   sqlplus username [/ as sysdba]
   Enter password: password
   ```

   For *`username`*, you can use the `SYS` or `SYSTEM` administrative user. At the prompt, enter the password that you set up during installation. If you use the `SYS` user, you must include `as sysdba` in the command line.

   SQL*Plus connects you to the default database instance (Microsoft Windows) or the database instance specified by the environment variables (Linux and UNIX systems).

To start SQL*Plus and connect to the database from the Start menu (Microsoft Windows):

1. Click **Start**.
2. From the Start menu, select **Programs** (or **All Programs**), then select *`Oracle_home`*, and then select **Application Development**.
3. Select **SQL*Plus**.
4. Enter values in the **User Name** and **Password** fields for the account you want to use to connect to the database. Then click **OK**.

   For the user name, you can use the `SYS` or `SYSTEM` administrative account, and you can use the password that you set up during installation. If you use the `SYS` account, enter the password in the **Password** field, then a space, and then the SQL clause `as sysdba`.

   When SQL*Plus loads, it issues the SQL prompt, which looks like this:

   ```
   SQL>
   ```
5. At the SQL prompt, you can enter statements that perform administrative tasks such as shutting down the database or creating a new user, or you can query, insert, update, and delete data.

   You can enter a single SQL statement on multiple lines. You must end each statement with a semicolon (;).

See Also:

- "About Administrative Accounts and Privileges" on page 7-4
- *SQL*Plus User's Guide and Reference*
- *Oracle Database SQL Language Reference*

About SQL Developer

SQL Developer is a graphical user interface for accessing your instance of Oracle Database. SQL Developer supports development in both the SQL and PL/SQL languages. It is available in the default installation of Oracle Database.

With SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, as well as create and save your own.

See Also: *Oracle Database 2 Day Developer's Guide*

Getting Started with Oracle Enterprise Manager: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Getting Started with Oracle Enterprise Manager OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_
dba/gettingstarted/gettingstarted.htm
```

4

Configuring the Network Environment

After installing Oracle Database, you have a fully functional database with a client/server network environment that has been minimally configured.

This chapter contains the following sections to help you completely configure your client/server network:

- Understanding Network Configuration
- Viewing Listener Configuration
- Starting and Stopping the Listener
- Connecting to an Oracle Database from a Client Computer
- Networking: Oracle By Example Series

Understanding Network Configuration

A **client** is any application that connects to Oracle Database to send or retrieve data. An Oracle Database client application can reside on any computer provided it has Oracle Database client software installed.

Oracle Net is a software layer that resides on the client computer and on the Oracle Database host computer. It establishes and maintains the connection between the client application and the database over a network, and exchanges messages between them using industry standard protocols.

For a client application and a database to communicate, the client application must be able to identify the database it wants to connect to, and the database must provide some sort of identification or address. You can use a service name to connect to a database. A **service name** is a logical representation of a database, which is the way a database is presented to clients. A single database can be presented as multiple services.

Using service names can provide location transparency such that the client application does not need to know the server's location. When the database is moved to another location, you need to reconfigure only Oracle Net. No changes are necessary to client applications.

See Also: *Oracle Database Net Services Administrator's Guide*

Oracle Net Listener Configuration

On the database server, the Oracle Net listener, commonly known as the listener, is a process that listens for client connection requests. It receives incoming client connection requests and manages the traffic of these requests to the database server.

The default listener configuration file is called `listener.ora`, and it is located in the *`Oracle_home`*`/network/admin` directory. The file contains a protocol address that identifies the database. This address defines the protocol the listener is listening on and any other protocol-specific information. For example, the listener could be configured to listen at the following protocol address:

```
(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=my-server) (PORT=1521)))
```

This example shows a TCP/IP address that specifies the host computer of the listener and a port number.

The `listener.ora` file is automatically configured during installation, but can be reconfigured with Oracle Enterprise Manager Database Control. For more information, see "Viewing Listener Configuration" on page 4-4 and "Starting and Stopping the Listener" on page 4-5.

Because all of the configuration parameters have default values, it is possible to start and use a listener without configuring it. This default listener has a name of `LISTENER`, supports no service names upon startup, and listens on the following TCP/IP protocol address:

```
(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)(PORT=1521))
```

An Oracle database registers with the listener within a minute or so of starting up. Supported service names, that is, the service names, or databases, to which the listener forwards client requests, can be configured in the `listener.ora` file, or this information can be dynamically registered with the listener. This dynamic registration feature is called service registration. The registration is performed by the process monitor (PMON) process—an instance background process—of each database instance that has the necessary configuration in the database initialization parameter file. Dynamic service registration does not require any configuration in the `listener.ora` file.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about configuring listeners

Connect Descriptors

The client uses a connect descriptor to specify the database to which it wants to connect. This connect descriptor contains a protocol and a database service name. A database can have multiple service names defined, so a specific service name must be specified for the connect descriptor. In a preconfigured database, there is only one service name, which defaults to the global database name.

The following example shows a connect descriptor that enables clients to connect to a database with service name `mydb.us.acme.com`:

```
(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=my-server) PORT=1521))
   (CONNECT_DATA=
      (SERVICE_NAME=mydb.us.acme.com)))
```

Connection Requests

Users initiate a connection request by providing a connect string. A connect string includes a user name and password, and a connect identifier. This connect identifier can be the connect descriptor itself, or a name that resolves to the connect descriptor

using mapping information stored in one or more repositories accessed with the naming methods described in "Naming Methods" on page 4-3. This name is referred to as a **net service name**.

Naming Methods

A **naming method** is a resolution method used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service.

Oracle Net provides support for the following naming methods:

- Easy Connect Naming
- Local Naming
- Directory Naming

Easy Connect Naming

The easy connect naming method enables clients to connect to an Oracle database by using only a TCP/IP connect string consisting of a host name and service name. The easy connect naming method requires no configuration. See "Connecting to an Oracle Database from a Client Computer" on page 4-6 for an example of easy connect naming.

Local Naming

The local naming method stores connect descriptors, identified by their net service names, in a configuration file on the client named `tnsnames.ora`. This file is located in the *`Oracle_home`*`/network/admin` directory. When you create a database using Oracle Database Configuration Assistant (DBCA), local naming is configured automatically.

Directory Naming

Directory naming resolves a database service, net service name, or net service alias to a connect descriptor stored in an LDAP-compliant directory server.

See Also: *Oracle Database Net Services Administrator's Guide*

Tools for Network Configuration

Oracle Database enables you to manage your network configuration with the following tools:

- Net Configuration Assistant
- Oracle Enterprise Manager
- Oracle Net Manager

Net Configuration Assistant

During a typical database installation, Net Configuration Assistant (NETCA) automatically configures a listener called `LISTENER` that has a TCP/IP listening protocol address for the database. If you do a custom installation, then NETCA prompts you to configure a listener name and protocol address of your choice.

Use NETCA for initial network configuration after database installation. Thereafter, you can use Oracle Enterprise Manager and Oracle Net Manager to configure and administer your networks.

Oracle Enterprise Manager

Oracle Enterprise Manager enables you to manage your server-side network configuration with a Listener page. The Listener page displays information such as the current status, Oracle home location, and host name.

See Also: "Viewing Listener Configuration" on page 4-4

Oracle Net Manager

Oracle Net Manager provides the same features as Oracle Enterprise Manager with the addition of configuring profiles.

See Also: *Oracle Database Net Services Administrator's Guide*

Viewing Listener Configuration

The listener runs on your database server computer and handles incoming client requests. With Oracle Enterprise Manager Database Control (Database Control), you can view the status of the listener, which is set to start automatically whenever your server computer is restarted. You can also view the listener status at the command line.

To view information about the listener with Database Control:

- On the Database Home page, click the **Listener** link under General.

  The Listener page appears, displaying the following information about the listener:

  - Current status
  - Availability over the last 24 hours
  - Alias
  - Version
  - Oracle home
  - Net address
  - Configuration file (`listener.ora`) location
  - Start time
  - Host

  For more information about a page at any time, click **Help** on that page.

To view information about the listener at the command line:

1. Open a command window.
2. (Linux and UNIX systems only) Set the required environment variables by running the script `oraenv` (for the Bourne, Korn, or Bash shells) or `coraenv` (for the C shell).

   These scripts are typically located in the `/usr/local/bin` directory. They may prompt you for the desired value of the `ORACLE_SID` environment variable. If so, supply the system ID (SID) that you chose when you installed Oracle Database. The default SID that the installer typically suggests is `orcl`.

3. (Linux and UNIX systems only) Either ensure that the *Oracle_home*/bin directory is in your PATH environment variable, or change the directory to *Oracle_home*/bin.
4. Enter the following command:

```
lsnrctl status
```

Starting and Stopping the Listener

The Oracle listener is set to start automatically whenever your server computer is restarted. However, when your system encounters unforeseen circumstances, or when you have manually stopped the listener, you can restart it with Database Control or at the command line.

To stop the listener with Database Control:

1. On the Database Home page, click the **Listener** link under General.

   The Listener page appears.
2. Click **Stop**.

To start the listener with Database Control:

1. On the Database Home page, click the **Listener** link under General.

   The Listener page appears.
2. Click **Start**.

> **Note:** If you have just stopped the listener and are still at the Database Control Listener page, then you will not see a Start button. You must go to the Database Home page, wait a minute or two, and then return to the Listener page.

To stop the listener at the command line:

1. Open a command window.
2. (Linux and UNIX only systems) Set the required environment variables by running the script oraenv (for the Bourne, Korn, or Bash shells) or coraenv (for the C shell).

   These scripts are typically located in the /usr/local/bin directory. They may prompt you for the desired value of the ORACLE_SID environment variable. If so, supply the system ID (SID) that you chose when you installed Oracle Database. The default SID that the installer typically suggests is orcl.
3. (Linux and UNIX systems only) Either ensure that the *Oracle_home*/bin directory is in your PATH environment variable, or change the directory to *Oracle_home*/bin.
4. Enter the following command:

```
lsnrctl stop
```

To start the listener at the command line:

1. Open a command window.

2. (Linux and UNIX only systems) Set the required environment variables by running the script `oraenv` (for the Bourne, Korn, or Bash shells) or `coraenv` (for the C shell).

   These scripts are typically located in the `/usr/local/bin` directory. They may prompt you for the desired value of the `ORACLE_SID` environment variable. If so, supply the system ID (SID) that you chose when you installed Oracle Database. The default SID that the installer typically suggests is `orcl`.

3. (Linux and UNIX systems only) Either ensure that the *`Oracle_home`*`/bin` directory is in your `PATH` environment variable, or change the directory to *`Oracle_home`*`/bin`.

4. Enter the following command:

   ```
   lsnrctl start
   ```

Connecting to an Oracle Database from a Client Computer

To connect to an Oracle database, you must first install Oracle Database client software on the client computer.

The easy connect naming method can eliminate the need for service name lookup in the `tnsnames.ora` files for TCP/IP environments. It enables clients to connect to an Oracle database by using only a TCP/IP connect identifier consisting of a host name and service name, as shown in the following example for Linux or UNIX systems:

```
sqlplus /nolog
SQL> CONNECT username@host/service_name
Enter password: password
```

This naming method provides TCP/IP connectivity to databases without first configuring Oracle Net Services.

To run SQL statements against the database, you use the SQL*Plus utility. On Microsoft Windows platforms, the environment is configured when the software is installed. On other platforms, you may have to configure environment variables such as `ORACLE_HOME` and `PATH` before you can start SQL*Plus.

To connect to an Oracle database from a Microsoft Windows client computer using easy connect naming:

1. On the desktop, click **Start**.
2. From the Start menu, select **Programs** (or **All Programs**), then select the folder that contains the installed Oracle software, for example, **Oracle - OracleDB11g_Home**.
3. Select **Application Development**, and then select **SQL*Plus**.

   When SQL*Plus starts, the Log On window appears first.

4. In the Log On window, enter the name and password of the user you want use to connect to the database. In the **Host String** field, supply the easy connect naming string for the target database, which is of the form:

   `host``/`*`service_name`*

   In this string, *`host`* is the name of the Oracle database computer and *`service_name`* is one of the services defined by the database. If there are no service names specifically defined, then you can use the name of the service that the database creates by default. This service name defaults to the global database name, a name

comprising the database name (`DB_NAME` parameter) and domain name (`DB_DOMAIN` parameter).

For example, to connect as the sample schema user `hr` to a database with the default database name `orcl` on a host named `main` at domain name `us.acme.com`, you would enter the following:

- User Name: **HR**
- Password: ***password***
- Host String: **main/orcl.us.acme.com**

5. The SQL*Plus window appears and displays the prompt:

```
SQL>
```

You can now issue SQL statements against the database.

Networking: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Networking OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/network/network.htm
```

5

Managing the Oracle Instance

This chapter provides background information about the Oracle instance and instructions about how to manage the instance.

This chapter contains the following sections:

- Overview of the Oracle Instance and Instance Management
- Shutting Down and Starting Up the Oracle Instance
- Viewing and Modifying Initialization Parameters
- Managing Memory
- Instances: Oracle By Example Series

Overview of the Oracle Instance and Instance Management

An Oracle database system consists of an Oracle database and an Oracle instance.

Oracle database—An Oracle database contains a set of operating system files that store data entered by users or applications, and structural information about the database itself called metadata. Information is stored persistently in these files.

Oracle instance—An Oracle instance (also known as a database instance) contains the set of Oracle Database background processes that operate on the stored data and the shared allocated memory that those processes use to do their work.

An instance must be started to read and write information to the database. Also, it is the instance that actually creates the database upon receipt of instructions from the Oracle Database Configuration Assistant (DBCA) utility or upon receipt of a SQL statement from the database administrator.

When the database instance is not available, your data is safe in the database, but it cannot be accessed by any user or application.

The properties of a database instance are specified using instance initialization parameters. When the instance is started, an initialization parameter file is read, and the instance is configured accordingly.

This section presents some of the concepts of an instance and its management. It contains the following topics:

- About Initialization Parameters
- About Background Processes
- About Server and Client Processes
- About Instance Memory Structure

- About Instance Startup and Shutdown

About Initialization Parameters

Managing an Oracle instance includes configuring parameters that affect the basic operation of the Oracle instance. These parameters are called initialization parameters. The Oracle instance reads initialization parameters from a file at startup.

During installation, when you select from one of the preconfigured database workloads available in DBCA, the initialization parameters are optimized for typical use in the environment that you specified. As the number of database users increases and the workload increases, you might need to alter some initialization parameters. You can make these changes using the Initialization Parameter page in Oracle Enterprise Manager Database Control (Database Control), or by using one of the advisors provided by Oracle Database, such as the Memory Advisor. See "Using the Memory Advisors" on page 10-27 for more information.

After being read from a file, initialization parameters are retained in memory, where the values for many of them can be changed dynamically. There are two types of files from which these parameters can be read: a *text initialization parameter file* and a *server parameter file*. The type of file used to start the instance determines if any dynamic initialization parameter changes you make are persistent across database shutdown and startup.

Server parameter file—The server parameter file is the preferred form of initialization parameter file, and is a binary file that can be written to and read by the database. It *must not* be edited manually. It is stored on the host computer on which Oracle Database is running. Any changes to it are persistent across database shutdown and startup. Changes are made when you use Database Control to modify one or more initialization parameters, or when Oracle Database itself makes changes for self-tuning purposes.

> **Note:** When changing an initialization parameter in the server parameter file, you can also specify that the in-memory value be changed, so that your change is reflected immediately in the current instance. If you do not change the in-memory value, the change does not take effect until you shut down and restart the database.

Text initialization parameter file—This type of initialization parameter file can be read by the Oracle instance, but it is not written to by the instance. You can change a text initialization parameter file with a text editor, but changes do not take effect until you shut down and restart the Oracle instance. When you start the instance with this type of file, you can still change many initialization parameters dynamically with Database Control, but only for the current instance. Unless you also edit the text initialization parameter file and make the same change, the change is lost when you shut down and restart the database.

When you create the database with DBCA (as described in "Using DBCA to Create and Configure a Database" on page 2-8), a server parameter file is created. This file is then used each time the database is started.

SQL statements exist to create a server parameter file from a text initialization file, to create a server parameter file from the current (in-memory) values of all initialization parameters, and to create a text initialization parameter file from a server parameter file.

See Also:

- "Viewing and Modifying Initialization Parameters" on page 5-10
- *Oracle Database Administrator's Guide* for information about the default name and location of the server parameter file, and for commands to create a server parameter file or text initialization parameter file

About Background Processes

The background processes of the Oracle instance manage memory structures, asynchronously perform I/O to write data to a file on a disk, and perform general maintenance tasks. The background processes consolidate functions that would otherwise be handled by multiple Oracle Database programs running for each user process. They monitor other Oracle Database processes to provide increased parallelism for better performance and reliability.

The background processes that are present depend on the features that are being used in the database. The most important background processes are described in Table 5–1.

Table 5–1 Oracle Database Background Processes

Background Process	Description
Database writer (DBW*n*)	The database writer writes modified blocks from the database buffer cache to the files on a disk. Oracle Database allows a maximum of 20 database writer processes.
Log writer (LGWR)	The log writer process writes redo log entries to a disk. Redo log entries are generated in the redo log buffer of the System Global Area (SGA) and the log writer process writes the redo log entries sequentially into an online redo log file.
Checkpoint	At specific times, all modified database buffers in the SGA are written to the datafiles by a database writer process (DBW*n*). This event is called a **checkpoint**. The checkpoint process signals DBW*n*, updates the datafiles and control files of the database, and records the time of this update.
System monitor (SMON)	The system monitor performs instance recovery when a failed instance is restarted.
Process monitor (PMON)	The process monitor performs a recovery when a user process fails. It cleans up the cache and frees resources that the failed process was using.
Archiver (ARC*n*)	Archiver processes copy the redo log files to archival storage when the log files are full or a log switch occurs. The database must be in archive log mode to run archive processes. For more information, see Chapter 9, "Performing Backup and Recovery".
Manageability monitor (MMON)	This process performs various management-related background tasks, for example: ■ Issuing alerts whenever a given metric violates its threshold value ■ Taking snapshots by spawning additional processes ■ Capturing statistical values for SQL objects that have been recently modified

About Server and Client Processes

In addition to background processes, Oracle Database creates server processes that handle the connection requests of user or client processes. A user connection is composed of two distinct pieces:

- A client program acting on behalf of the user, such as Oracle Enterprise Manager, SQL*Plus, or an application
- A server process that handles the connection to the database on behalf of the client program, and that performs much of the work for the client program, such as parsing and running SQL statements, and retrieving and returning results to the client program

Server processes can be either dedicated or shared. When server processes are dedicated, Oracle Database is considered to be running in **dedicated server mode**. When server processes are shared, Oracle Database is considered to be running in **shared server mode**. In dedicated server mode, each client process has its own server process. Although a dedicated server process is good for long-running queries and administrative tasks, an idle process or too many dedicated processes can result in an inefficient use of resources.

Using shared server mode eliminates the need for a dedicated server process for each connection. In shared server mode, each client connection request is sent to a *dispatcher*, which is a process that is responsible for routing requests from connected user processes to a pool of shared server processes and returning the responses back to the appropriate user processes. An idle shared server process picks up a request from a common queue, which means a small number of shared servers can perform the same amount of processing as many dedicated servers. Because the amount of memory required for each user is relatively small, less memory and process management are required, and more users can be supported.

Shared server mode is more efficient at supporting multiple client programs making frequent short-running queries.

About Instance Memory Structure

The sizes of the instance memory structures affect the performance of Oracle Database and are controlled by initialization parameters.

Upon installation, you can choose to let the database manage memory for you automatically, or you can configure some degree of manual memory management. If you configure manual memory management, Oracle Database provides advisors to help you determine appropriate values for memory parameters. For both manual and automatic memory management, Oracle Database sends alerts that identify memory sizing problems that require your attention.

The following sections describe the two important memory areas that you can monitor and size:

- System Global Area
- Program Global Area

See Also:

- "Managing Memory" on page 5-12
- "Memory" on page 2-13 for information about memory management options during installation
- *Oracle Database Concepts* for more information about Oracle Database memory architecture

System Global Area

The System Global Area (SGA) is a shared memory area that contains data and control information for the instance. Multiple users can share data within this memory area and avoid repeated, time-consuming access from a physical disk.

The SGA has several components, as listed in Table 5–2.

Table 5–2 SGA Components

Component	Description
Buffer cache	Before any data stored in the database can be queried or modified, it must be read from a disk and stored in memory in the buffer cache. All user processes connected to the database share access to the buffer cache. For optimal performance, the buffer cache should be large enough to avoid frequent disk read and write operations.
Shared pool	The shared pool caches information that is shared among users: ■ SQL statements that can be reused ■ Information from the data dictionary such as user account data, table and index descriptions, and privileges ■ Stored procedures, which are executable code that is stored in the database
Redo log buffer	This buffer improves performance by caching redo information until it can be written to the physical redo log files stored on disk. Redo information and redo log files are discussed in "About Online Redo Log Files" on page 6-3.
Large pool	This optional area is used to buffer large I/O requests for various server processes.
Java pool	The Java pool is an area of memory that is used for all session-specific Java code and data within the Java Virtual Machine (JVM).
Streams pool	The Streams pool is an area of memory that is used by the Oracle Streams feature. For more information about Oracle Streams, see *Oracle Streams Concepts and Administration*.
Result cache	The result cache buffers query results. If a query is run that already has results in the result cache, the database returns results from the result cache instead of rerunning the query. This speeds up the execution of frequently run queries.

Program Global Area

A Program Global Area (PGA) is a memory area used by a single Oracle Database server process. When you connect to an Oracle database instance, you create a *session* that uses a server process for communication between the client and database instance. Each server process has its own PGA.

The PGA is used to process SQL statements and to hold logon and other session information. A large part of the PGA is dedicated to *SQL work areas*, which are working memory areas for sorts and other SQL operations.

The amount of PGA memory used and the contents of the PGA depend on whether the instance is running in dedicated server or shared server mode. See *Oracle Database Concepts* for details. See also "About Server and Client Processes" on page 5-4 for more information about dedicated server and shared server modes.

The total memory used by all individual PGAs is known as the **total instance PGA memory**, and the collection of individual PGAs is referred to as the **total instance PGA**, or just **instance PGA**. With Oracle Enterprise Manager Database Control, you set the size of the instance PGA, not individual PGAs.

About Instance Startup and Shutdown

After installation, the Oracle instance is started, and the database is open for access by users with database accounts. At some point, you may want to shut down and restart the instance. This section describes the startup and shutdown processes.

The phrases "starting up and shutting down the Oracle instance" are often used interchangeably with "starting up and shutting down the database."

This section contains the following topics:

- About Administration Privileges for Startup and Shutdown
- About Instance Startup
- About Instance Shutdown

See Also:

- "Shutting Down and Starting Up the Oracle Instance" on page 5-7

About Administration Privileges for Startup and Shutdown

To start up or shut down the Oracle instance, you must connect to the instance with a special connection privilege. There are two of these privileges: `SYSDBA` for fully empowered database administrators and `SYSOPER` for users who operate the database, but have no privileges to access user objects.

When you create an Oracle database, there are two primary administrative user accounts that are automatically created: `SYS` and `SYSTEM`. Both of these users have full database administration privileges, but initially, only user `SYS` or `SYSTEM` can connect with the `SYSOPER` privilege. Therefore, until you grant the `SYSOPER` privilege to other users, you must connect to the Oracle instance as user `SYS` or `SYSTEM` to start up and shut down the instance. When connecting (logging in) as user `SYS`, you must always specify that you are connecting `AS SYSDBA`.

See Also:

- "About Administrative Accounts and Privileges" on page 7-4 for more information about the `SYS` and `SYSTEM` users, and the `SYSDBA` privilege

About Instance Startup

When you start the Oracle instance, you typically start it such that the database is *open* and ready for user connections. However, there are situations in which you may want to start the instance with the database *mounted*, but not open. An instance can also be

started without the database either mounted or open. Thus, there are three stages to starting an instance:

1. You start the instance using one of the following methods:
   - Using Oracle Enterprise Manager. See "Shutting Down and Starting Up the Oracle Instance" on page 5-7.
   - Using the SQL*Plus `STARTUP` statement. See *Oracle Database Administrator's Guide*.
   - On Microsoft Windows, using the Services program in Control Panel to start the Oracle Database services. See "Shutting Down and Starting Up Using the Windows Services Program" on page 5-9.

   The instance reads the initialization parameter file, allocates SGA memory, and starts the background processes.
2. If you mount the database, the instance opens the database control file, but does not open the datafiles. The database is now considered to be in the *mount* state. This state enables you to perform certain administrative functions that cannot be performed when other users are accessing the database. An example of such a function is enabling or disabling redo log archiving. See "About Archived Redo Log Files" on page 6-4 for information about redo log archiving.
3. If you open the database, the instance is first mounted, and then it opens the redo log files and datafiles for the database. The database is now open and available for all user access.

The default startup behavior is to complete the three stages in sequence, transparently. Unless you explicitly specify otherwise, the instance is started and the database is mounted and opened.

About Instance Shutdown

Instance shutdown is the reverse of instance startup. Shutdown goes through these three stages:

1. You shut down the database using one of the methods discussed in "About Instance Startup" on page 5-6. Any modified data blocks cached in the SGA that have not been written to a disk are written. The contents of the redo log buffer are written to a disk, and then the redo log buffer is cleared. A checkpoint is performed on the datafiles and their headers are marked current as of the time the database was closed. Datafiles and log files are closed. Users can no longer access the database.
2. The Oracle instance dismounts the database and updates relevant entries in the control file to record a clean shutdown. The control file is closed. The database is now closed and dismounted, and only the instance remains.
3. The Oracle instance stops the background processes and deallocates the shared memory used by the SGA.

Shutting Down and Starting Up the Oracle Instance

This section provides instructions about two methods you can use to start up or shut down the Oracle instance:

- Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control
- Shutting Down and Starting Up Using the Windows Services Program

See Also:

- "About Instance Startup and Shutdown" on page 5-6

Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control

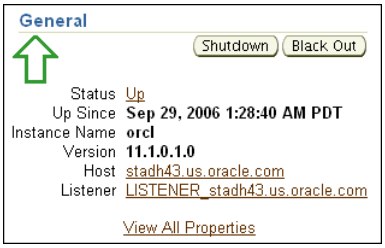
You can shut down and start up the Oracle instance from the Database Home page of Oracle Enterprise Manager Database Control (Database Control).

To shut down and start up the Oracle instance using Database Control:

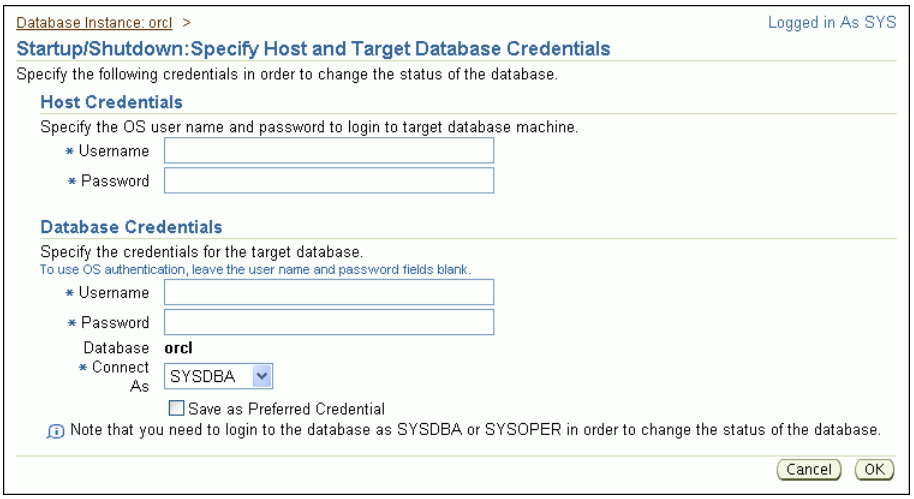
1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. In the General section of the Database Home page, click **Shutdown**.



   The Startup/Shutdown Credentials page appears.



3. Enter credentials as follows:

   a. Enter the host computer credentials of the user who installed Oracle Database, or of any user who is authorized to use SQL*Plus.

   b. Enter the database credentials consisting of the user name `SYS` and the password that you assigned to `SYS` during the installation.

   c. In the **Connect As** list, choose the value `SYSOPER`.

4. (Optional) Select the **Save as Preferred Credential** option if you want these credentials to be automatically filled in for you the next time that this page appears.

5. Click **OK**.

   A confirmation page appears.

6. Click **Yes**.

   The Startup/Shutdown: Activity Information page appears, informing you that the database is being shut down.

7. After a short period of time, click **Refresh** to be returned to the Database Home page.

   The Database Home page indicates that the database instance status is Down.

8. Click **Startup** to start up the database.

   The Startup/Shutdown Credential page appears again.

9. Enter the credentials, and then click **OK**.

   A confirmation page appears.

10. Click **Yes**.

    The Startup/Shutdown: Activity Information page appears, indicating that the database is being started up. When startup is complete, the Login page appears.

11. Log in to the database (and to Database Control).

    The Database Home page appears indicating that the database instance status is Up.

Shutting Down and Starting Up Using the Windows Services Program

On Microsoft Windows, you can also start up and shut down your Oracle database using the Services program in Control Panel. You must start or stop the following services:

- `OracleService`*`SID`*, which is your Oracle instance.
- `Oracle`*`Oracle_home`*`TNSListener`, which is your listener. The listener is required for clients to connect to your database.
- `OracleDBConsole`*`SID`*, which enables clients to connect to Oracle Enterprise Manager Database Control.

In the preceding service names, *`SID`* refers to the system identifier for the instance and *`Oracle_home`* refers to the Oracle home name.

To start or stop Oracle Database services:

1. Do one of the following:
   - On Windows XP and Windows Server 2003, click **Start**, and then select **Control Panel**.
   - On Windows 2000 Server, click **Start**, select **Settings**, and then select **Control Panel**.

The Control Panel window opens.

2. Double-click the **Administrative Tools** icon, and then double-click the **Services** icon.

   The Services window opens, displaying all Windows services that are available on your system.

3. Locate the Oracle Database services listed at the beginning of this section. For example, if your SID is `orcl`, then locate the following services:
   - `OracleServiceORCL`
   - `OracleOraDb11g_home1TNSListener`
   - `OracleDBConsoleorcl`
4. Start or stop all three services, using the following steps for each service:
   a. Select the service name.
   b. In the **Action** menu, click **Start** or **Stop**.

Viewing and Modifying Initialization Parameters

This section provides instructions about viewing the initialization parameter settings for your database and modifying these parameters. Three scenarios are explained for modifying initialization parameters:

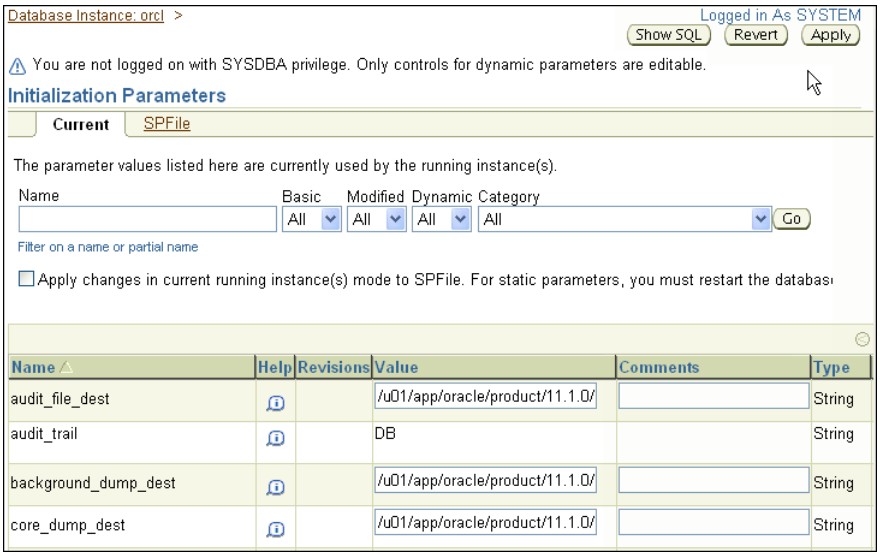
- Modifying initialization parameters for the currently running instance only. The modifications are lost when the database is shut down and restarted.
- Modifying initialization parameters for the currently running instance, and also recording the modifications in the server parameter file, so that the modifications persist when the database is restarted.
- Modifying initialization parameters only in the server parameter file. The current instance is not affected and changes take effect only when the database is next restarted.

> **Note:** These three scenarios correspond to using the `SCOPE=MEMORY`, `SCOPE=BOTH`, and `SCOPE=SPFILE` clauses of the `ALTER SYSTEM` SQL statement, respectively, when you use the `ALTER SYSTEM` SQL statement to change initialization parameters.

To view or modify initialization parameters:

1. At the top of the Database Home page, click the **Server** link to view the Server subpage.
2. Under Database Configuration, click **All Initialization Parameters**.

   Oracle Enterprise Manager Database Control (Database Control) displays the Initialization Parameters page.



The Initialization Parameters page has two subpages:

- **Current**—This subpage (the default) displays all initialization parameter values that are currently active (in memory) for the Oracle instance.
- **SPFile**—This subpage displays initialization parameter settings in the server parameter file. This subpage is present only when the current instance started up with a server parameter file. The file location is displayed at the top of the subpage.

See the online Help for the Initialization Parameters page for field descriptions.

3. (Optional) On either subpage, reduce the number of initialization parameters displayed by doing one or both of the following, and then clicking **Go**:
   - In the **Name** field, enter text.
   - Select from one or more of the lists next to the **Name** field.

   For example, to view only initialization parameters that have the text "dest" anywhere in the parameter name, enter `dest` in the **Name** field, and then click **Go**.
4. To modify one or more initialization parameters for the currently running instance only, with the modifications being lost when the instance is restarted, complete the following steps:
   a. On the Current subpage, in the **Value** column, enter new values for the initialization parameters.

   Note: If the Value column cannot be written to for a particular initialization parameter, it means that this parameter is not dynamic—that is, it cannot be changed for the current instance.

   b. Ensure that **Apply changes in current running instance(s) mode to SPFile** is not selected.
   c. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
   d. Click **Apply**.

A confirmation message appears.

5. To modify initialization parameters for the currently running instance, and also record the modifications in the server parameter file that will persist when the database is restarted, complete the following steps:
   a. On the Current subpage, in the **Value** column, enter new values for the initialization parameters.
   b. Select **Apply changes in current running instance(s) mode to SPFile**.
   c. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
   d. Click **Apply**.

      A confirmation message appears.
6. To modify initialization parameters in the server parameter file only, such that the current instance is not affected and changes take effect only when the database is next restarted, complete the following steps:
   a. Click the **SPFile** link to view the SPFile subpage.
   b. (Optional) Reduce the number of entries in the initialization parameter list as described in Step 3.
   c. In the **Value** column, enter new values for the initialization parameters.
   d. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
   e. Click **Apply**.

      A confirmation message appears.

Note: Changes to initialization parameters are recorded in the alert log as `ALTER SYSTEM` statements. See *Oracle Database Administrator's Guide* for information about the alert log.

See Also:

- "About Initialization Parameters" on page 5-2

Managing Memory

This section provides background information about managing memory for the Oracle instance, and includes instructions about how to adjust the memory allocation for the Oracle instance. It contains the following topics:

- About Memory Management
- Enabling Automatic Memory Management
- Modifying Memory Settings – Automatic Memory Management
- Modifying Memory Settings – Automatic Shared Memory Management
- Modifying Memory Settings – Automatic PGA Memory Management

See Also:

- "About Instance Memory Structure" on page 5-4

About Memory Management

Memory management involves maintaining optimal sizes for the Oracle instance memory structures as demands on the database change. The memory that must be managed is the System Global Area (SGA) memory and the instance Program Global Area (instance PGA) memory. The instance PGA memory is the collection of memory allocations for all individual PGAs.

Beginning with Release 11*g*, Oracle Database can manage the SGA memory and instance PGA memory completely automatically. You designate only the total memory size to be used by the instance, and Oracle Database dynamically exchanges memory between the SGA and the instance PGA as needed to meet processing demands. This capability is referred to as *automatic memory management*. In this memory management mode, the database also dynamically tunes the sizes of the individual SGA components and the sizes of the individual PGAs.

If you want to exercise more direct control over the sizes of the SGA and instance PGA, you can use the Memory Advisors page of Database Control to disable automatic memory management and enable *automatic shared memory management*. When you do this, you also implicitly enable *automatic PGA memory management*. With automatic shared memory management, you set target and maximum sizes for the SGA. The database then tunes the total size of the SGA to your designated target, and dynamically tunes the sizes of all SGA components. With automatic PGA memory management, you set a target size for the instance PGA. The database then tunes the size of the instance PGA to your target, and dynamically tunes the sizes of individual PGAs.

If you want complete control of individual SGA component sizes, you can use the Memory Advisors page of Database Control to disable both automatic memory management and automatic shared memory management. This effectively enables *manual shared memory management*. In this mode, you set the sizes of several individual SGA components, thereby determining the overall SGA size. You then manually tune these individual SGA components on an ongoing basis. This mode is intended for experienced DBAs only. Note that in this mode, automatic PGA memory management remains enabled.

Note: Although it is possible to disable automatic PGA memory management, it is not recommended, and cannot be done using the Memory Advisors page of Database Control.

Table 5–3 summarizes the various memory management modes that you can set using the Memory Advisors page of Database Control.

Table 5–3 Oracle Database Memory Management Modes

Memory Management Mode	You Set	Oracle Database Automatically Tunes
Automatic memory management	■ Total memory size for this instance ■ (Optional) Maximum memory size for this instance	■ Total SGA size ■ SGA component sizes ■ Instance PGA size ■ Individual PGA sizes
Automatic shared memory management and automatic PGA memory management (Automatic memory management disabled)	■ SGA target size ■ (Optional) SGA maximum size ■ Instance PGA target size	■ SGA component sizes ■ Individual PGA sizes
Manual shared memory management and automatic PGA memory management (Automatic memory management and automatic shared memory management disabled)	■ Shared pool size ■ Buffer cache size ■ Java pool size ■ Large pool size ■ Instance PGA target size	■ Individual PGA sizes

Note: Automatic Memory Management is not available on all platforms. See *Oracle Database Administrator's Guide* for more information about supported platforms.

If you choose the basic installation option when you install the database, automatic memory management is enabled. If you choose advanced installation, DBCA enables you to select from the three memory management modes. Oracle recommends that you enable automatic memory management.

Whichever memory management mode you choose, you may have occasion to adjust memory settings as demands on the database or on its host computer change. Reasons why you adjust memory settings include the following:

- You receive a memory-related alert or error message.
- You receive a memory-related recommendation from Automatic Database Diagnostic Monitor (ADDM).
- You want to change the amount of memory allocated to accommodate future growth in memory demand.

You can use a *memory advisor* to help you adjust memory sizes. See "Modifying Memory Settings – Automatic Shared Memory Management" on page 5-18 for an example of using a memory advisor.

See Also:

- "Enabling Automatic Memory Management" on page 5-15
- "Modifying Memory Settings – Automatic Memory Management" on page 5-16
- "Modifying Memory Settings – Automatic PGA Memory Management" on page 5-20
- "Memory" on page 2-13 for information about setting memory parameters during installation

Enabling Automatic Memory Management

If you did not enable automatic memory management when you installed and configured your database, Oracle recommends that you do so after installation, unless you are an experienced DBA with specific reasons to manually tune memory sizes. With automatic memory management, the Oracle instance dynamically tunes all memory components to optimize performance as the workload changes.

To enable automatic memory management:

1. Go to the Database Home page, logging in as user `SYS`.

   See "Accessing the Database Home Page" on page 3-4.

   Note: You must log in as user `SYS` to change the maximum memory size, which is a static initialization parameter.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

   The Memory Advisors page appears with the SGA subtab displayed by default.
4. If the Maximum SGA Size field already contains a positive number that is greater than or equal to the desired total amount of memory to allocate to the database, skip to Step 13.
5. In the **Maximum SGA Size (MB)** field, enter the maximum permissible size for database memory.

   Use the sum of the current sizes of the SGA and instance PGA as a guideline, and optionally add some room for growth.
6. Click **Apply**.

   A confirmation page appears, indicating that you must restart the database.
7. Click **Yes**.

   The Restart Database: Specify Host and Target Database Credentials page appears.
8. Enter host credentials and database credentials.

   Supply host credentials for any operating system user that has execute permission on SQL*Plus. Supply database credentials for a database user with the `SYSDBA` privilege (such as the user `SYS`).
9. Click **OK**.

   The Restart Database: Confirmation page appears.

10. Click **Yes** to proceed with database restart.
11. Wait a few minutes, and then click **Refresh**.

    The Database Home page appears.
12. Return to the Memory Advisors page.
13. Next to Automatic Memory Management **Disabled**, click **Enable**.

    The Enable Automatic Memory Management page appears.
14. In the field titled **Total Memory Size for Automatic Memory Management**, enter the desired amount of memory to allocate to the database, and then click **OK**.

    A confirmation message appears.

See Also:

- "About Memory Management" on page 5-13

Modifying Memory Settings – Automatic Memory Management

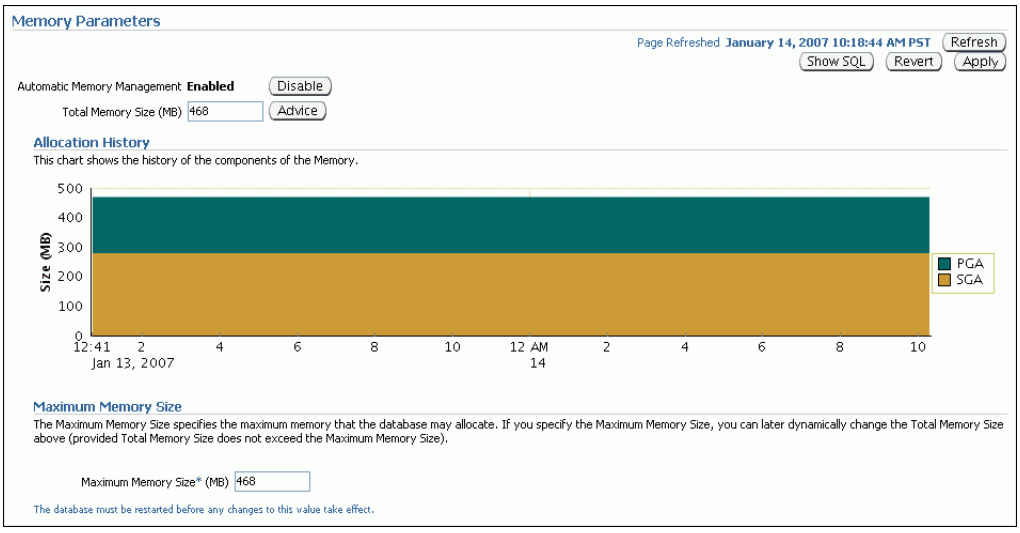
Modifying memory settings for automatic memory management involves using the Memory Advisors page of Database Control to change the total memory size setting. The procedure for changing this setting differs depending on whether or not your desired new setting exceeds the current setting for maximum memory size. If the new setting exceeds the maximum memory size setting, you must first increase the maximum memory size accordingly.

To modify total memory size (new value does not exceed maximum memory size):

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

   The Memory Parameters page appears.



4. Do one of the following:

- In the **Total Memory Size** field, enter a new value that is less than or equal to the value in the current Maximum Memory Size field.
- To use the Memory Advisor to help you choose a new size, click **Advice**.

  The Memory Size Advice page appears.

  Click a point on the curve to change the total memory size. The new size is displayed below the graph. Continue clicking the curve until the desired value is displayed, and then click **OK**.

5. On the Memory Advisors page, click **Apply**.

   A confirmation message appears.

To modify total memory size (new value exceeds maximum memory size):

1. Go to the Database Home page, logging in as user `SYS`.

   See "Accessing the Database Home Page" on page 3-4.

   Note: You must log in as user `SYS` to change the maximum memory size, which is a static initialization parameter.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

   The Memory Advisors page appears.

4. In the **Maximum Memory Size (MB)** field, enter a new value for the maximum total memory to allocate to the database.

   This value must be greater than or equal to the desired new value for total memory size.

5. Click **Apply**.

   A confirmation page appears, indicating that you must restart the database.

6. Click **Yes**.

   The Restart Database page appears.

7. Enter host credentials and database credentials.

   Supply host credentials for any user that has execute permission on SQL*Plus. Supply database credentials for a database user with the `SYSDBA` privilege (such as user `SYS`).

8. Click **OK**.

   The Restart Database: Confirmation page appears.

9. Click **Yes** to proceed with database restart.
10. Wait a few minutes, and then click **Refresh**.

    The Database Home page appears.

11. Return to the Memory Advisors page.
12. Do one of the following:

    - In the **Total Memory Size** field, enter a new value that is less than or equal to the value in the current Maximum Memory Size field.

- To use the Memory Advisor to help you choose a new size, click **Advice**.

  The Memory Size Advice page appears.

  Click a point on the curve to change the total memory size. The new size is displayed below the graph. Continue clicking the curve until the desired value is displayed, and then click **OK**.

13. On the Memory Advisors page, click **Apply**.

    A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- "Using the Memory Advisors" on page 10-27
- "About Memory Management" on page 5-13

Modifying Memory Settings – Automatic Shared Memory Management

Modifying memory settings for automatic shared memory management involves using the Memory Advisors page of Database Control to change the total SGA size. This section assumes that automatic memory management is disabled, that automatic shared memory management is enabled, and that the desired new value for the total SGA size does not exceed the current value for maximum SGA size.

Note: If you want a new total SGA size that exceeds the current value for maximum SGA size, you must first log in as user `SYS`, change the maximum SGA size, and restart the database. A similar procedure is described in "To modify total memory size (new value exceeds maximum memory size):" on page 5-17.

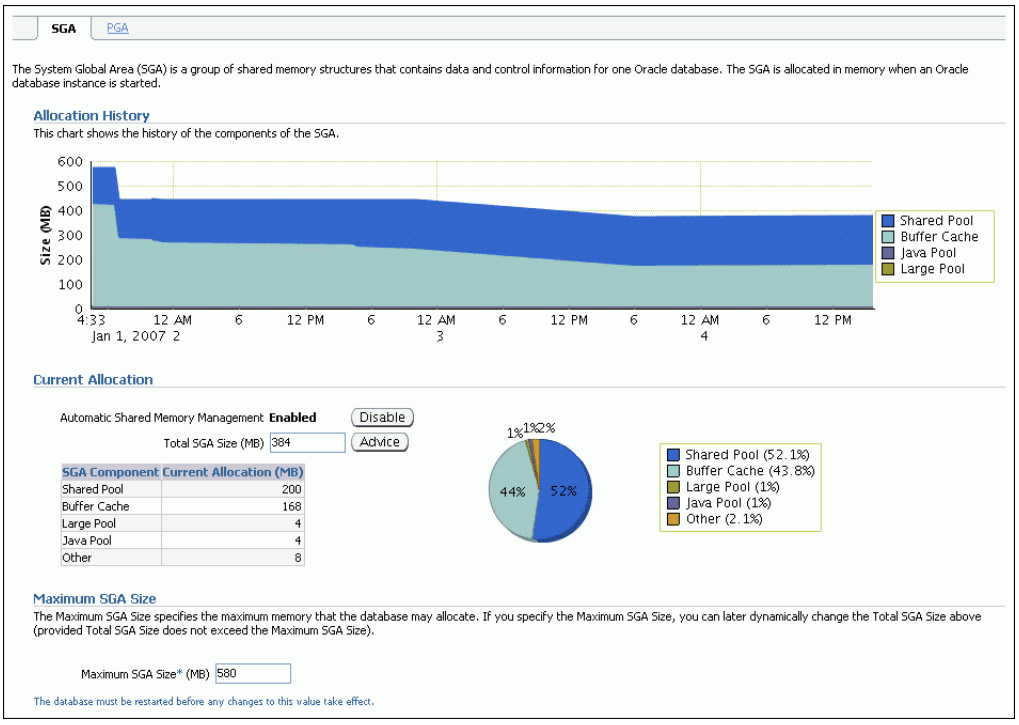
To modify the total SGA memory size:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

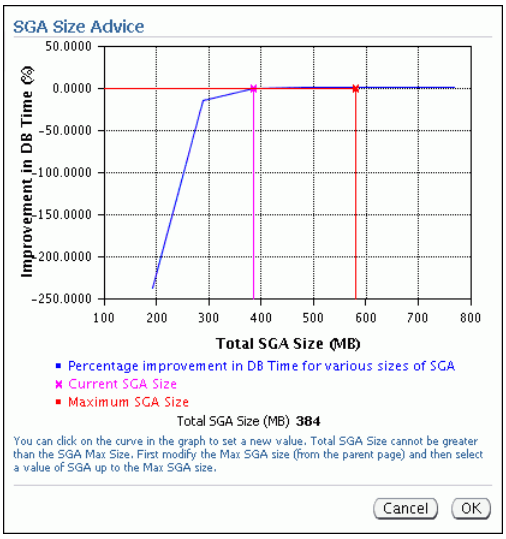
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

   The Memory Advisors page appears, with the SGA subpage shown.



4. In the Current Allocation section, do one of the following:
   - In the **Total SGA Size (MB)** field, enter a new total size for the SGA.
   - To use the SGA Advisor to help you choose a new SGA size, click **Advice**.

     The SGA Size Advice window opens.



   Click a point on the curve to change the total SGA size. The new SGA size is displayed below the graph. Continue clicking the curve until the desired value is displayed, and then click **OK**.

   This example graph shows that at the current time, increasing the total SGA size is not expected to provide any gain in performance.

5. On the Memory Advisors page, click **Apply**.

A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- "Using the Memory Advisors" on page 10-27
- "About Memory Management" on page 5-13

Modifying Memory Settings – Automatic PGA Memory Management

Modifying memory settings for automatic PGA memory management involves using the Memory Advisors page of Database Control to modify the instance PGA size. This section assumes that automatic memory management is disabled, and that automatic PGA memory management is enabled.

Note: When you disable automatic memory management, automatic PGA memory management is enabled by default.

To modify the instance PGA size:

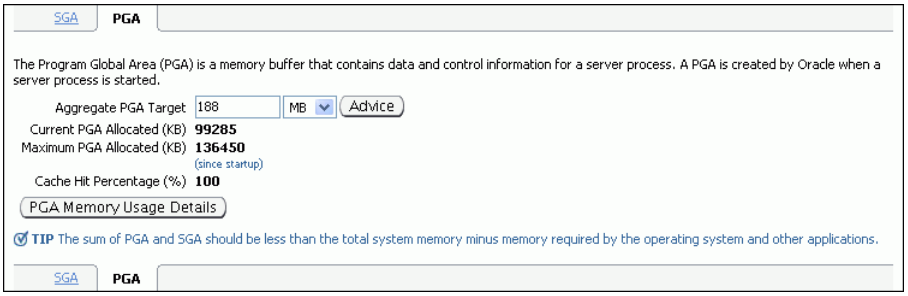
1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

   The Memory Advisors page appears, with the SGA subpage shown.

4. Click **PGA** to view the PGA subpage.



5. Do one of the following:
   - In the **Aggregate PGA Target** field, enter a new size for the instance PGA.
   - To use the PGA Advisor to help you choose a new instance PGA size, click **Advice**.

     The PGA Aggregate Target Advice window appears. Click a point on the curve to change the instance PGA size, and then click **OK**.

   See *Oracle Database Performance Tuning Guide* for information about sizing the instance PGA.

6. Click **Apply**.

   A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- "About Memory Management" on page 5-13

Instances: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter, and includes annotated screenshots.

To view the Instances OBE, in your browser, enter the following URL:

`http://www.oracle.com/technology/obe/11gr1_2day_dba/instance/instance.htm`

6

Managing Database Storage Structures

This chapter discusses using Oracle Enterprise Manager Database Control (Database Control) to view and manage the storage structures of your database. This chapter contains the following sections:

- About Database Storage Structures
- Viewing Database Storage Structure Information
- Performing Common Database Storage Tasks
- Managing the Redo Log
- Managing Undo Data
- Storage: Oracle By Example Series

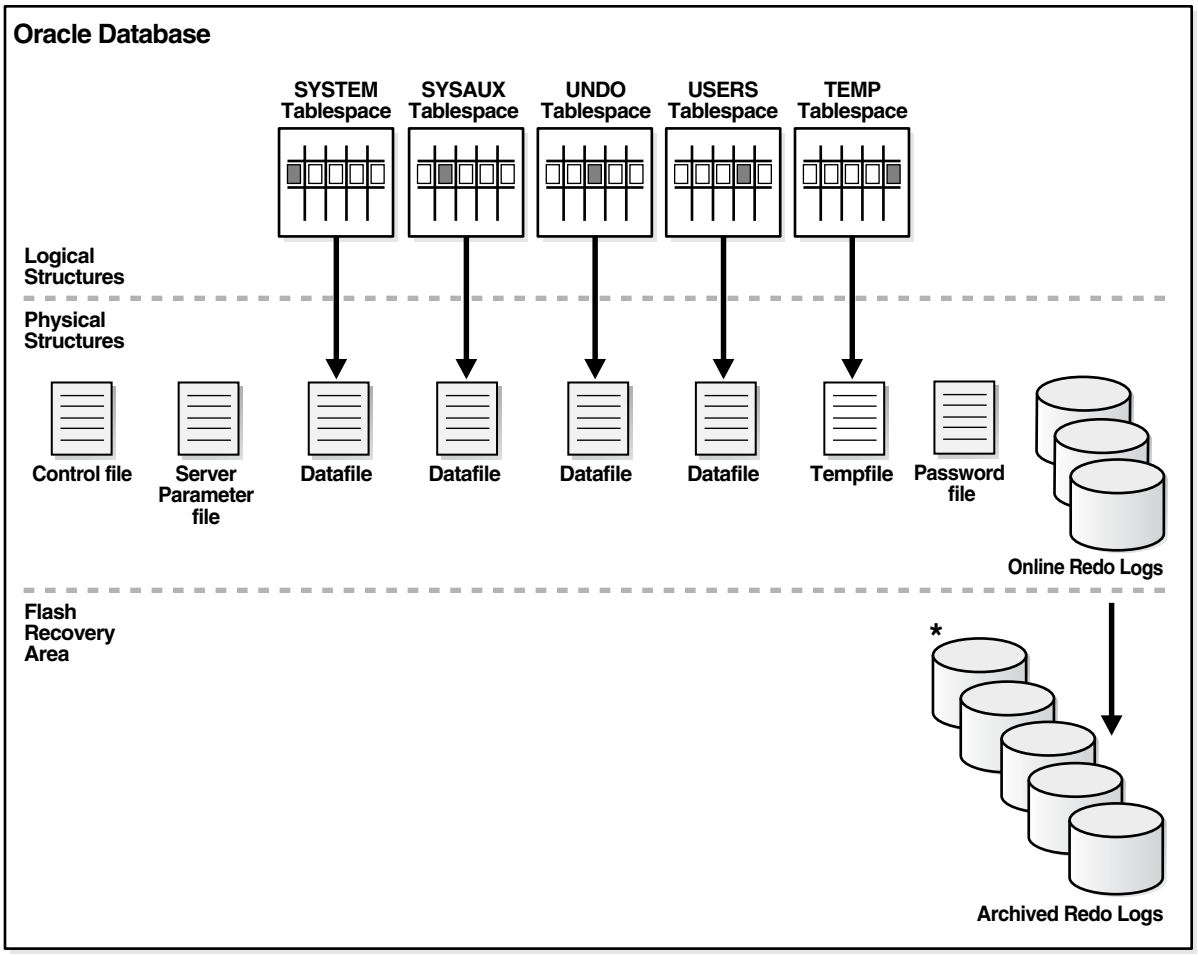
About Database Storage Structures

An Oracle database is made up of physical and logical structures. Physical structures are those that can be seen and operated on from the operating system, such as the physical files that store data on a disk.

Logical structures are created and recognized by Oracle Database and are not known to the operating system. The primary logical structure in a database, a tablespace, contains physical files. The applications developer or user may be aware of the logical structure, but is not usually aware of this physical structure. The DBA must understand the relationship between the physical and logical structures of a database.

Figure 6–1 on page 6-2 shows the relationships between logical and physical structures. This figure also shows recovery-related structures that are optionally kept in the flash recovery area. See "Flash Recovery Area" on page 9-3 for more information.

Figure 6–1 Oracle Database Storage Structures

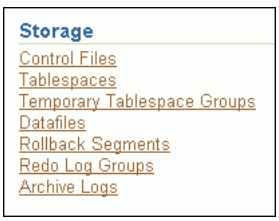


* Archived Redo Logs present only after turning on log archiving (ARCHIVELOG mode)

Oracle Database can automate much of the management of its structure. Oracle Enterprise Manager Database Control (Database Control) provides a Web-based graphical user interface (GUI) to enable easier management and monitoring of your database.

To view a database storage structure, go to the Storage section of the Server subpage. You can click the links shown in Figure 6–2 to access the storage pages.

Figure 6–2 Storage Options



This section provides background information about the various database storage structures. It contains the following topics:

- About Control Files

- About Online Redo Log Files
- About Archived Redo Log Files
- About Rollback Segments
- About Datafiles
- About Tablespaces
- About Other Storage Structures

See Also:

- "Viewing Database Storage Structure Information" on page 6-10

About Control Files

A control file tracks the physical components of the database. It is the root file that the database uses to find all the other files used by the database. Because of the importance of the control file, Oracle recommends that the control file be *multiplexed*. In other words, the control file should have multiple identical copies. For databases created with Oracle Database Configuration Assistant (DBCA), three copies of the control file are automatically created and kept synchronized with each other.

If any control file fails, then your database becomes unavailable. As long as you have a control file copy, however, you can shut down your database and re-create the failed control file from the copy, then restart your database. Another option is to delete the failed control file from the `CONTROL_FILES` initialization parameter and restart your database using the remaining control files.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about control files

About Online Redo Log Files

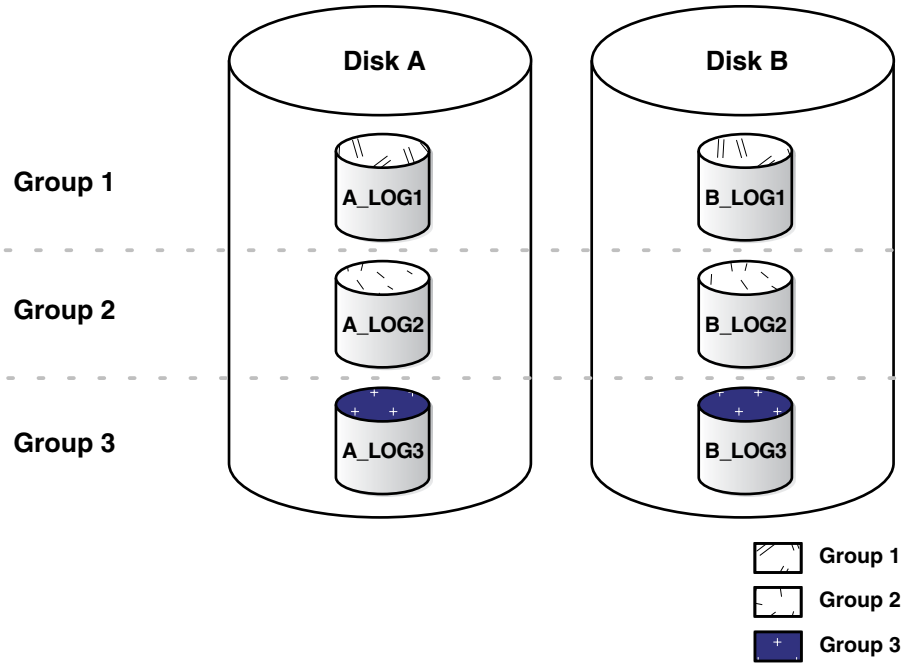
Every Oracle database has a set of two or more online redo log files. The set of redo log files is collectively known as the redo log for the database. A redo log is made up of redo entries, which are also called *redo records*.

The redo log stores a copy of the changes made to data. If a failure requires a datafile to be restored from backup, then the recent data changes that are missing from the restored datafile can be obtained from the redo log, so work is never lost. The redo log is used to recover a database after hardware, software, or media failure. To protect against a failure involving the redo log itself, Oracle Database can *multiplex* the redo log so that two or more identical copies of the online redo log can be maintained on different disks.

The redo log for a database consists of groups of redo log files. A group consists of a redo log file and its multiplexed copies. Each identical copy is considered to be a member of that group. Each group is defined by a number, such as Group 1.

Figure 6–3 on page 6-4 shows the configuration of a database that has three redo log groups and two members in each group. For each group, the members are stored on separate disks for maximum availability. For example, the members of Group 1 are the redo log files `A_LOG1` and `B_LOG1`.

Figure 6–3 Online Redo Log Groups and Their Members



The database log writer process (LGWR) writes redo records from the memory buffer to a redo log group until the log files in that group reach their storage size limit, or until you request a log switch operation. The LGWR process then writes to the next log group. The LGWR process performs this action in a circular fashion so that the oldest group is overwritten by the most recent redo records.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about redo logs
- Table 5–1, " Oracle Database Background Processes" on page 5-3 for more information about the LGWR process
- "Switching a Log File" on page 6-24

About Archived Redo Log Files

When you archive your redo log, you copy the redo log files to another location before they are overwritten. These copied files are referred to as archived redo log files. You can archive to multiple locations, including a standby database.

These archived redo log files extend the amount of redo data that can be saved and are used for recovery. Archived redo log files are required to recover a backup of the database from the time of the backup to the current time. Archiving can be either enabled or disabled for the database, but Oracle strongly recommends that you enable archiving. Oracle also recommends that you configure the database to write archived redo log files to the flash recovery area.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about archived redo log files
- "Configuring Recovery Settings" on page 9-6 for information about enabling redo log archiving
- "Flash Recovery Area" on page 9-3 for background information about the Flash Recovery Area

About Rollback Segments

Rollback segments were database structures used to track undo information for the database in earlier releases of Oracle Database. Now, the preferred way of managing undo information is with the undo tablespace. For more information, see "Managing Undo Data" on page 6-25.

Note: Oracle Database uses a `SYSTEM` rollback segment for performing system transactions. It is created automatically when the database is created, and is always brought online at instance startup. It is located in the `SYSTEM` tablespace. You are not required to perform any operations to manage the `SYSTEM` rollback segment.

About Datafiles

Datafiles are the operating system files that store the data within the database. The data is written to these files in an Oracle proprietary format that cannot be read by other programs. **Tempfiles** are a special class of datafiles that are associated only with temporary tablespaces.

Datafiles can be broken down into the following components:

- Segment

  A **segment** contains a specific type of database object. For example, a table is stored in a table segment, and an index is stored in an index segment. A datafile can contain many segments.

- Extent

  An **extent** is a contiguous set of data blocks within a segment. Oracle Database allocates space for segments in units of one extent. When the existing extents of a segment are full, the database allocates another extent for that segment.

- Data block

  A **data block**, also called a *database block*, is the smallest unit of I/O to database storage. An extent consists of several contiguous data blocks. The database uses a default block size at database creation.

  After the database has been created, it is not possible to change the default block size without re-creating the database. It is possible, however, to create a tablespace with a block size different than the default block size.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about datafiles
- *Oracle Database Concepts* for more information about segments, extents, and blocks

About Tablespaces

A database is divided into logical storage units called **tablespaces**, which group together related logical structures (such as tables, views, and other database objects). For example, all application objects can be grouped into a single tablespace to simplify maintenance operations.

A tablespace consists of one or more physical datafiles. Database objects assigned to a tablespace are stored in the physical datafiles of that tablespace.

When you create an Oracle database, some tablespaces already exist, such as `SYSTEM` and `USERS`.

Tablespaces provide a means to physically locate data on storage. When you define the datafiles that make up a tablespace, you specify a storage location for these files. For example, you might specify a datafile location for a certain tablespace as a designated host directory (implying a certain disk volume) or designated Automatic Storage Management disk group. Any schema objects assigned to that tablespace then get located in the specified storage location. Tablespaces also provide a unit of backup and recovery. The backup and recovery features of Oracle Database enable you to back up or recover at the tablespace level.

Table 6–1 describes some of the tablespaces included in the database.

Table 6–1 Tablespaces and Descriptions

Tablespace	Description
`EXAMPLE`	This tablespace contains the sample schemas that are included with Oracle Database. The sample schemas provide a common platform for examples. Oracle documentation and educational materials contain examples based on the sample schemas.
`SYSTEM`	This tablespace is automatically created at database creation. Oracle Database uses it to manage the database. It contains the data dictionary, which is the central set of tables and views used as a read-only reference for a particular database. It also contains various tables and views that contain administrative information about the database. These are all contained in the `SYS` schema, and can be accessed only by the `SYS` user or other administrative users with the required privilege.
`SYSAUX`	This is an auxiliary tablespace to the `SYSTEM` tablespace. Some components and products that used the `SYSTEM` tablespace or their own tablespaces in releases prior to Oracle Database 10*g* now use the `SYSAUX` tablespace. Using `SYSAUX` reduces the load on the `SYSTEM` tablespace and reduces maintenance because there are fewer tablespaces to monitor and maintain. Every Oracle Database 10*g* or later database release must have a `SYSAUX` tablespace. Components that use `SYSAUX` as their default tablespace during installation include Automatic Workload Repository, Oracle Streams, Oracle Text, and Database Control Repository. For more information, see *Oracle Database Administrator's Guide*.

Table 6–1 (Cont.) Tablespaces and Descriptions

Tablespace	Description
TEMP	This tablespace stores temporary data generated when processing SQL statements. For example, this tablespace would be used for query sorting. Every database should have a temporary tablespace that is assigned to users as their temporary tablespace. In the preconfigured database, the TEMP tablespace is specified as the default temporary tablespace. If no temporary tablespace is specified when a user account is created, then Oracle Database assigns this tablespace to the user.
UNDOTBS1	This is the undo tablespace used by the database to store undo information. See "Managing Undo Data" on page 6-25 to understand how an Oracle database uses the undo tablespace. Every database must have an undo tablespace.
USERS	This tablespace is used to store permanent user objects and data. Similar to the TEMP tablespace, every database should have a tablespace for permanent user data that is assigned to users. Otherwise, user objects will be created in the SYSTEM tablespace, which is not good practice. In the preconfigured database, USERS is designated as the default tablespace for all new users.

You can create new tablespaces to support your user and application data requirements. During tablespace creation, you set the following parameters:

- Locally Managed Tablespaces Compared to Dictionary-Managed Tablespaces
- Tablespace Type
- Tablespace Status
- Autoextend Tablespace

Locally Managed Tablespaces Compared to Dictionary-Managed Tablespaces

Space management within a tablespace involves keeping track of available (free) and used space, so that space is allocated efficiently during data insertion and deletion. Oracle recommends creating locally managed tablespaces rather than dictionary-managed tablespaces. Dictionary-managed tablespaces is an older space management technique and not as efficient as locally managed tablespaces.

Locally managed tablespaces keep the space allocation information within the tablespace, not in the data dictionary, thus offering better performance. By default, Oracle Database sets all newly created tablespaces to be locally managed with automatic segment management, a feature that further improves performance.

Tablespace Type

There are three types of tablespaces:

- Permanent

  Oracle Database uses permanent tablespaces to store permanent data, such as system data. You use permanent tablespaces to store your user and application data. Each user is assigned a default permanent tablespace.

- Undo

  A database running in automatic undo management mode transparently creates and manages undo data in the undo tablespace. Oracle Database uses undo data to roll back transactions, to provide read consistency, to help with database recovery, and to enable features such as Oracle Flashback Query.

Even though you can create more than one undo tablespace, only one can be active. If you want to switch the undo tablespace used by the database instance, then you can create a new one and instruct the database to use it instead. The undo tablespace no longer in use can then be removed from the database (or *dropped*).

- Temporary

  Temporary tablespaces are used for storing temporary data, as would be created when SQL statements perform sort operations. An Oracle database gets a temporary tablespace when the database is created. You would create another temporary tablespace if you were creating a temporary tablespace group. Under typical circumstances, you do not need to create additional temporary tablespaces. If you have an extremely large database, then you might configure additional temporary tablespaces.

  The physical files that make up a temporary tablespace are called tempfiles, as opposed to datafiles.

  The `TEMP` tablespace is typically used as the default temporary tablespace for users who are not explicitly assigned a temporary tablespace.

  See Also:

  - *Oracle Database Administrator's Guide* to learn more about temporary tablespaces
  - *Oracle Database Concepts* for more information about read-consistency and Oracle Flashback Query

Tablespace Status

You can set tablespace status as follows:

- Read Write

  Users can read and write to the tablespace after it is created. This is the default.

- Read Only

  If the tablespace is created Read Only, then the tablespace cannot be written to until its status is changed to Read Write. It is unlikely that you would create a Read Only tablespace, but you might change it to that status after you have written data to it that you do not want modified.

- Offline

  If the tablespace is created Offline, then no users can access it. It is unlikely that you will create an Offline tablespace, but later you might change its status to Offline to perform maintenance on its datafiles.

Autoextend Tablespace

You can set a tablespace to automatically extend itself by a specified amount when it reaches its size limit. If you do not enable autoextend, you are alerted when the tablespace reaches its critical or warning threshold size. The critical and warning threshold parameters have default values that you can change at any time. These parameters also cause alerts to be generated for autoextending tablespaces that are approaching their specified size limit. You can respond to size alerts by manually increasing the tablespace size. You do so by increasing the size of one or more of the tablespace datafiles or by adding another datafile to the tablespace.

Note: Although it is common to refer to tablespaces as autoextending, automatic extension is a datafile property, not a tablespace property. That is, when you create the datafiles that make up a tablespace, you indicate whether or not these datafiles automatically extend. A tablespace that has autoextending datafiles is considered to be an autoextending tablespace. You can specify a maximum size for an autoextending datafile.

About Other Storage Structures

Other storage structures that can exist in an Oracle database include the initialization parameter file, the password file, and backup files.

This section contains the following topics:

- Initialization Parameter File
- Password File
- Backup Files

Initialization Parameter File

Initialization parameters are used by the Oracle instance at startup to determine the run-time properties and resources for the database. Some parameters can be set or modified while the database is running. Other initialization parameters require the database to be restarted for the changes to take effect.

See Also:

- Chapter 5, "Managing the Oracle Instance" for more information about initialization parameters and the initialization parameter file

Password File

A database can use a password file to authenticate administrative users with `SYSDBA` or `SYSOPER` connection privileges. These privileges enable a DBA to start up and shut down the database and perform other high-level administrative tasks. This password file is outside of the database itself, thereby enabling the authentication of a DBA when the database is not yet started. (A DBA must authenticate before starting the database.)

When you invoke DBCA as part of the Oracle Database installation process, DBCA creates a password file with one entry: the `SYS` user. Granting `SYSDBA` to a user adds that user to the password file automatically.

Note: Oracle Database can also use *operating system authentication* to authenticate users with the `SYSDBA` or `SYSOPER` privileges.

See Also:

- *Oracle Database Administrator's Guide* for more information about password files and operating system authentication
- "SYSDBA and SYSOPER System Privileges" on page 7-5

Backup Files

Backup files are not technically database files, but are copies of the database in some form that can be used to recover the database if a failure causes loss of data.

See Also:

- Chapter 9, "Performing Backup and Recovery" for more information about backup files

Viewing Database Storage Structure Information

This section provides instructions about viewing information about the various database storage structures with Oracle Enterprise Manager Database Control (Database Control). It contains the following topics:

- Viewing Control File Information
- Viewing Online Redo Log File Information
- Viewing Archive Log Information
- Viewing Datafile Information
- Viewing Tablespace Information

See Also:

- "About Database Storage Structures" on page 6-1

Viewing Control File Information

You use Database Control to view location and status information about control files.

To view control file information:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Control Files**.

   The Control Files page appears, showing the General subpage. This page shows whether or not your database has a multiplexed control file. The Advanced and Record Section subpages give you more detailed information about your control files.

   For more information about a page, at any time, click **Help**.

Viewing Online Redo Log File Information

You use Database Control to view status and multiplexing information about online redo log files.

To view redo log file information:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Redo Log Groups**.

The Redo Log Groups page appears. This page shows the attributes of the redo log groups for your database.

When a redo log group contains only one member, it is not multiplexed. Note the Status attribute for the redo log groups. The group with status Current is the one currently being written to disk by the log writer.

For more information about a page, at any time, click **Help**.

4. (Optional) In the **Group** column, click a group number to view information about redo log group members.

Viewing Archive Log Information

You use Database Control to view status information about archived redo log files.

> **Note:** Archived redo log files do not exist until you set the database in `ARCHIVELOG` mode.

To view archive log information:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Archive Logs**.

   The Archive Logs page appears.

   For more information about a page, at any time, click **Help**.

Viewing Datafile Information

You use Database Control to view status and location information about datafiles.

To view datafile information:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Datafiles**.

   The Datafiles page appears.

   For more information about a page, at any time, click **Help**.

4. (Optional) Select a datafile, and then click **View** to display more attributes for the datafile. You can also click the name of a tablespace in the **Tablespace** column to display the attributes of the tablespace associated with the datafile.

Viewing Tablespace Information

You use Database Control to view configuration, size, and status information about tablespaces.

To view tablespace information:

1. Go to the Database Home page.

See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Tablespaces**.

   The Tablespaces page appears.

   For more information about a page, at any time, click **Help**.

Performing Common Database Storage Tasks

As data is added to your database, the tablespace requirements for your database change. As a DBA, you must understand how to perform the following tasks to effectively manage the tablespaces and database storage:

- Creating a Tablespace
- Modifying a Tablespace
- Dropping a Tablespace
- Reclaiming Unused Space

Creating a Tablespace

You can create additional tablespaces to store user data, so that not all data is stored in the `USERS` tablespace. The following are some reasons to create additional tablespaces:

- For certain users, groups of users, or applications, it may be convenient to keep all application data in a separate tablespace or set of tablespaces for backup and recovery or maintenance reasons. For example, suppose you must recover all application data from backup due to a hardware or software failure, and you want to perform an offline recovery. If the application data is kept in a separate tablespace, you can take just that tablespace offline and recover it, without affecting the operation of other database applications.
- Some applications, such as those with large partitioned tables, may benefit from distributing data across multiple tablespaces. This approach allows the optimal use of the available storage because frequently accessed data can be placed on high performance disks, and infrequently retrieved data can be placed on less expensive storage.

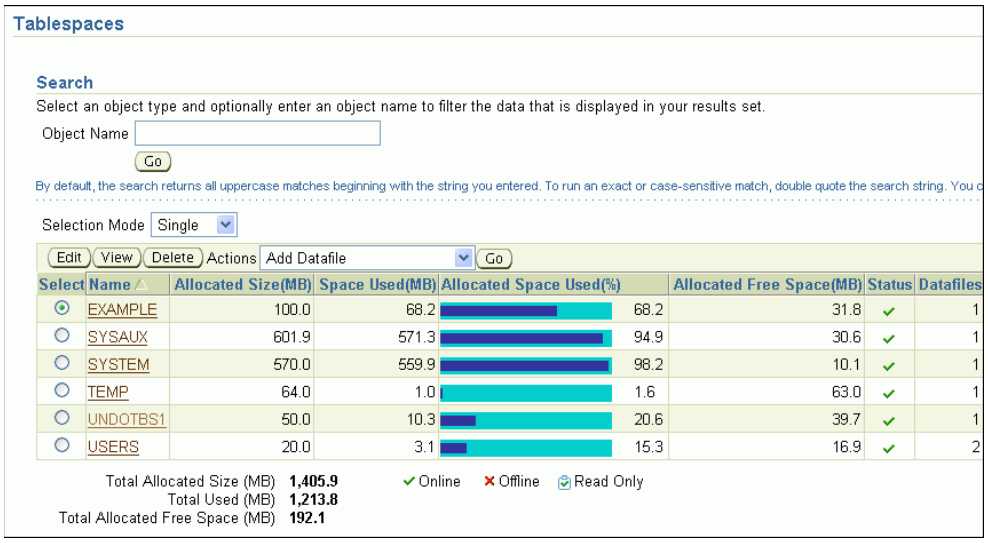
To create a tablespace:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.
2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Tablespaces**.

   The Tablespaces page appears, as shown in Figure 6–4.

Figure 6–4 Tablespaces Page



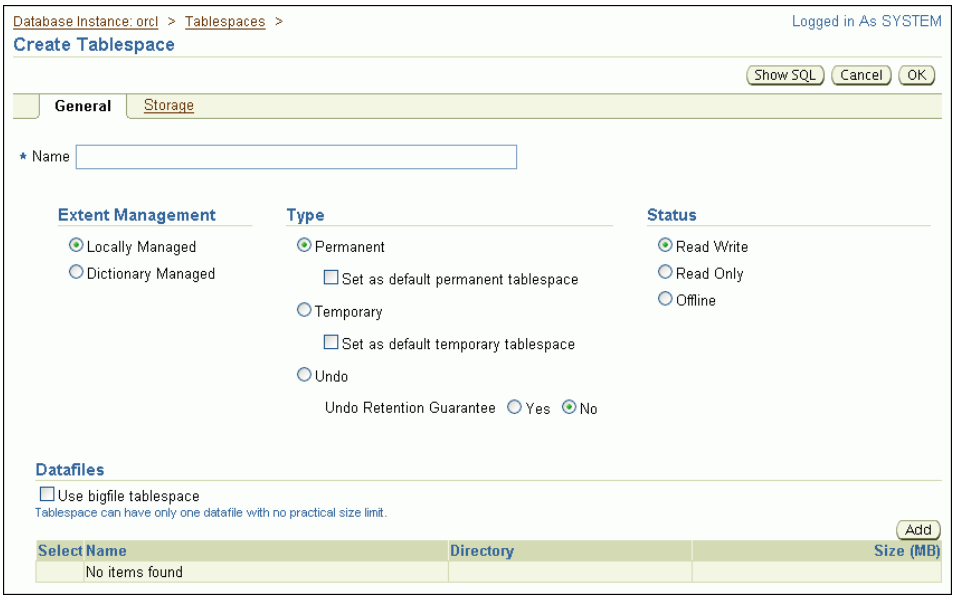
For more information about a page, at any time, click **Help**.

4. Do one of the following:

   a. To create a new tablespace, click **Create** (not shown in the screenshot).

   b. To create a tablespace that is similar to an existing tablespace, in the **Select** column, select the tablespace whose attributes you want to reproduce. In the Actions list, select **Create Like** and then click **Go**.

   The Create Tablespace page appears, showing the General subpage.



5. In the **Name** field, enter a name for the tablespace.
6. In the Extent Management section, select **Locally Managed**.

   See "Locally Managed Tablespaces Compared to Dictionary-Managed Tablespaces" on page 6-7.
7. In the Type section, select **Permanent**.

   See "Tablespace Type" on page 6-7.
8. In the Status section, select **Read Write**.

   See "Tablespace Status" on page 6-8.
9. In the Datafiles section, click **Add** to add one or more datafiles to the tablespace.

   The Add Datafile page appears.

   Note: If you select **Use bigfile tablespace**, then the tablespace can have only one datafile. Bigfile tablespaces are used with very large databases that use Automatic Storage Management or other logical volume managers that support striping, RAID, and dynamically extensible logical volumes.

   ---

10. In the **File Name** field, enter a file name. Accept the **File Directory** and **File Size** defaults.
11. In the Storage section, complete the following steps:
    a. Select **Automatically extend datafile when full (AUTOEXTEND)**.
    b. Set a suitable increment, such as 10 MB.

       This is the amount of disk space that is added to the datafile when it needs more storage space.
    c. For Maximum File Size, do one of the following, depending on available storage:
       – Select **Unlimited** to permit the file to increase without limits.

– Select **Value**, and then enter a value in KB, MB, GB, or TB.

12. Click **Continue**.

    The Create Tablespace page returns.

13. (Optional) Toward the top of the page, click the **Storage** link to view the Storage subpage. Examine all of the default settings on this subpage.

    See the online Help for more information about these settings.

14. Click **OK** to add the tablespace.

    The Tablespaces page returns, showing the newly created tablespace.

See Also:

- "About Tablespaces" on page 6-6

Modifying a Tablespace

You can use Oracle Enterprise Manager Database Control (Database Control), to modify a tablespace. For example, you can extend it (by increasing datafile sizes or adding another datafile), set it to automatically extend, change its space usage alert thresholds, or change its status to Offline. When you get a critical or warning alert, you might need to extend a tablespace (if the alert is related to space available) or take it offline to recover it (if the alert is related to corrupted data or other serious errors).

This section contains the following topics:

- Setting a Tablespace to Automatically Extend
- Changing Space Usage Alert Thresholds for a Tablespace
- Taking a Tablespace Offline

See Also:

- "About Tablespaces" on page 6-6

Setting a Tablespace to Automatically Extend

This section explains how to set a tablespace to automatically extend when it reaches its size limit. The following instructions assume that the tablespace was previously not an autoextending tablespace.

To set a tablespace to automatically extend:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.

3. In the Storage section, click **Tablespaces**.

   The Tablespaces page appears.

   See Figure 6–4 on page 6-13.

   For more information about a page, at any time, click **Help**.

4. Select the tablespace for which you want to enable autoextend, and then click **Edit**.

   The Edit Tablespace page appears.

5. Select the first datafile associated with the tablespace, and then click **Edit**.

The Edit Datafile page appears.

6. In the Storage section, complete the following steps:
   a. Select **Automatically extend datafile when full (AUTOEXTEND)**.
   b. Set a suitable increment, such as 10 MB.

      This is the amount of disk space that is added to the datafile when it needs more storage space.
   c. For Maximum File Size, do one of the following, depending on available storage:
      – Select **Unlimited** to permit the file to increase without limits.
      – Select **Value**, and then enter a value in KB, MB, GB, or TB.
7. Click **Continue**.

   The Edit Tablespace page returns.
8. Repeat Step 5 through Step 7 for each additional datafile associated with the tablespace.
9. Click **Apply**.

   A confirmation message appears.

Changing Space Usage Alert Thresholds for a Tablespace

You receive an alert in Database Control when a space usage threshold for a tablespace is reached. There are two types of space usage alerts that you can enable: **warning**, for when tablespace space is somewhat low, and **critical**, for when the tablespace is almost completely full and action must be taken immediately.

For both warning and critical alerts, there are two ways to specify alert thresholds:

- By space used (%)

  When space used becomes greater than or equal to a percentage of total space, an alert is issued.
- By free space (MB)

  When remaining space falls below an amount (in MB), an alert is issued.

  Free-space thresholds are more useful for large tablespaces. For example, for a 10 TB tablespace, setting the percentage full critical alert to as high as 99 percent means that the database would issue an alert when there is still 100 GB of free space remaining. In many cases, 100 GB remaining would not be a critical situation, and the alert would not be useful. For this tablespace, it might be better to use a free-space threshold, which you could set to issue a critical alert when 5 GB of free space remains.

For both warning and critical alerts for a tablespace, you can enable either the space used threshold or the free-space threshold, or you can enable both thresholds.

To change space usage alert thresholds for tablespaces:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.
2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Tablespaces**.

The Tablespaces page appears.

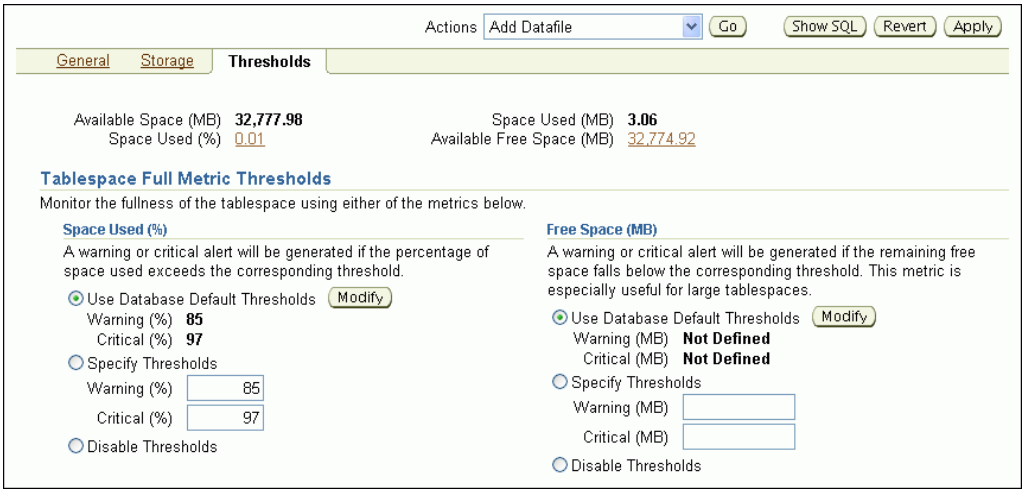
See Figure 6–4 on page 6-13.

For more information about a page, at any time, click **Help**.

4. Select the tablespace whose threshold you want to change, and then click **Edit**.

   The Edit Tablespace page appears, showing the General subpage.

5. Click **Thresholds** to display the Thresholds subpage.



6. In the Space Used (%) section, do one of the following:
   - Accept the default thresholds.
   - Select **Specify Thresholds**, and then enter a **Warning (%)** threshold and a **Critical (%)** threshold.
   - Select **Disable Thresholds** to disable the percentage full thresholds.
7. In the Free Space (MB) section, do one of the following:
   - Accept the default thresholds.
   - Select **Specify Thresholds**, and then enter a **Warning (MB)** threshold and a **Critical (MB)** threshold.
   - Select **Disable Thresholds** to disable the threshold for free space remaining.
8. Click **Apply**.

   A confirmation message appears.

Taking a Tablespace Offline

You may want to take a tablespace offline for any of the following reasons:

- To make a portion of the database unavailable while still allowing access to the remainder of the database
- To make an application and its group of tables temporarily unavailable while updating or maintaining the application
- To perform an offline tablespace backup (even though a tablespace can be backed up while online and in use)
- To recover a tablespace after a hardware or software failure

- To rename or relocate tablespace datafiles

To take a tablespace offline:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Tablespaces**.

   The Tablespaces page appears.

   See Figure 6–4 on page 6-13.

   For more information about a page, at any time, click **Help**.

4. Select the tablespace that you want to take offline, and then click **Edit**.

   The Edit Tablespace page appears.

5. In the Status section, select **Offline**, make a selection from the Offline Mode list, and then click **Apply**.

   For more information about the selections on this page, click **Help**.

   A confirmation message appears.

Note: To bring the tablespace back online, return to this page, and then, under the Status section, click **Read Write**.

See Also:

- *Oracle Database Administrator's Guide* for more information about taking tablespaces offline and for information about renaming or relocating datafiles.

Dropping a Tablespace

After a tablespace has been dropped (deleted), the objects and data in it are no longer available. To recover them can be a time-consuming process. Oracle recommends performing a backup before and after dropping a tablespace.

To drop a tablespace:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Tablespaces**.

   The Tablespaces page appears.

   For more information about a page, at any time, click **Help**.

4. Select the tablespace that you want to drop, and then click **Delete**.

   Database Control asks for confirmation that you want to delete the tablespace and delete the associated datafiles from the disk.

5. Click **Yes** to remove the tablespace.

See Also:

- "About Tablespaces" on page 6-6

Reclaiming Unused Space

This section provides background information about reclaimable unused space in the database, and provides instructions about how to reclaim this space. It contains the following topics:

- About Reclaimable Unused Space
- About the Segment Advisor
- Viewing Segment Advisor Recommendations and Reclaiming Space
- Running the Segment Advisor Manually

About Reclaimable Unused Space

Over time, performing insert, update, and delete operations (also referred to as DML operations) on objects within a tablespace can create pockets of empty space that individually are not big enough to be reused. Collectively, these pockets can waste large amounts of space. The sparsely populated objects that result can suffer performance degradation during queries and DML operations.

Oracle Database enables you reclaim this empty space with the following two online operations—that is, operations that do not block queries or DML against affected objects while the operations are proceeding:

- Online segment shrink

  Online segment shrink operates on table segments and on the segments of the dependent objects of the table, such as indexes and partitions. For each segment, data is compacted to the front of the segment. Free space can either be returned to the tablespace or kept in the segment for future insert operations. Online segment shrink is permitted only on segments in a locally managed tablespace with automatic segment space management. Online segment shrink is an in-place operation; no additional free space in the tablespace is required.

- Online table redefinition (also known as *reorganization*)

  Reorganization relocates the table and its dependent objects in a different part of the tablespace. This operation has the desirable side effect of compacting the data for those objects. For this operation to succeed, however, the tablespace must already have free space equal to the size of the table and its dependent objects. Reorganization is the only permitted operation for dictionary-managed tablespaces or for locally managed tablespaces with manual segment space management.

The Segment Advisor generates recommendations for shrinking or reorganizing segments that have a significant amount of reclaimable unused space.

See Also:

- *Oracle Database Administrator's Guide* for more information about online segment shrink and online redefinition of tables

About the Segment Advisor

The Segment Advisor identifies database objects that have unused space that you can reclaim. It performs its analysis by examining usage and growth statistics and by

sampling the data in the object. By default, it is configured to run automatically at regular intervals, during all maintenance windows (time periods). You can also run the Segment Advisor manually.

The regularly scheduled Segment Advisor task is known as the Automatic Segment Advisor. Results from the advisor are summarized on the Space Summary section of the Database Home page as Segment Advisor Recommendations. See Figure 6–5 on page 6-21.

During each run of the Automatic Segment Advisor, only a subset of the segments in the database are analyzed. If you believe that particular segments may have reclaimable unused space, yet those segments do not appear among the Segment Advisor recommendations, the Automatic Segment Advisor may not have chosen them for analysis. You can run the Segment Advisor manually on those segments at any time. See *Oracle Database Administrator's Guide* for information about how the Automatic Segment Advisor selects tablespaces and segments for analysis.

Segment Advisor Recommendations Segment Advisor recommendations are grouped by tablespace. Only tablespace names appear on the Segment Advisor Recommendations page in Database Control. If a tablespace contains any segments for which a shrink operation or reorganization is recommended, the tablespace appears on the Segment Advisor Recommendations page. This page estimates the amount of reclaimable space for each tablespace. If you request recommendation details for a tablespace, Database Control displays the segments in that tablespace for which recommendations were generated.

See Figure 6–6 on page 6-21 and Figure 6–7 on page 6-22.

A recommendation for a segment can be either a shrink operation or a reorganization operation. If you created the tablespace for a segment as locally managed with automatic segment space management, which is the default, the Segment Advisor recommends shrinking, if the segment is not one of the few segment types that are not eligible for shrinking. An example of such as segment is a table with function-based indexes. If the segment is not eligible for online segment shrink, or if its tablespace is dictionary-managed or was created with manual segment space management, the Segment Advisor recommends reorganization.

See Also:

- *Oracle Database Administrator's Guide* for more information about maintenance windows

Viewing Segment Advisor Recommendations and Reclaiming Space

Each Segment Advisor recommendation includes a button that you can click to start the space reclamation process. Although space reclamation is an online process, it can consume significant resources, so it is recommended that you reclaim space during off-peak hours.

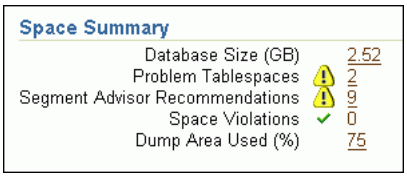
To view Segment Advisor recommendations and reclaim space:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

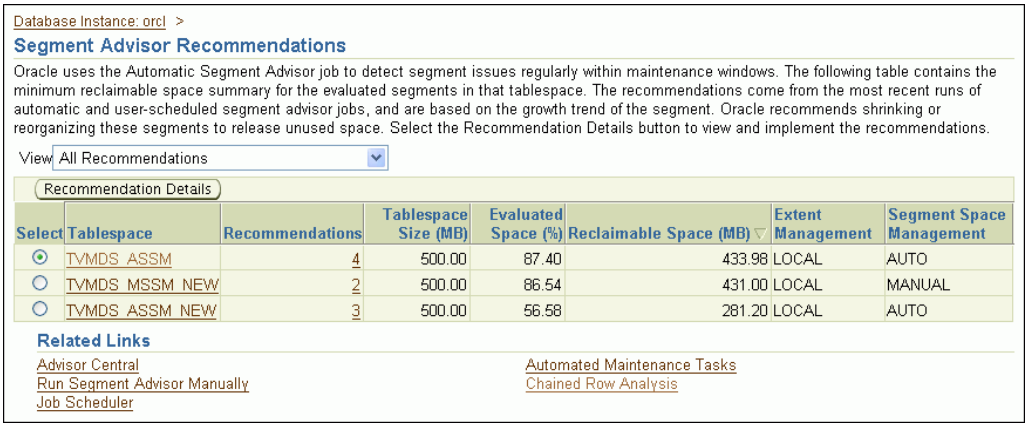
2. In the Space Summary section, click the numeric link adjacent to Segment Advisor Recommendations.

Figure 6–5 Home Page Space Summary Section



The Segment Advisor Recommendations page appears, showing recommendations from the most recent automatic and manual runs of the Segment Advisor.

Figure 6–6 Segment Advisor Recommendations Page



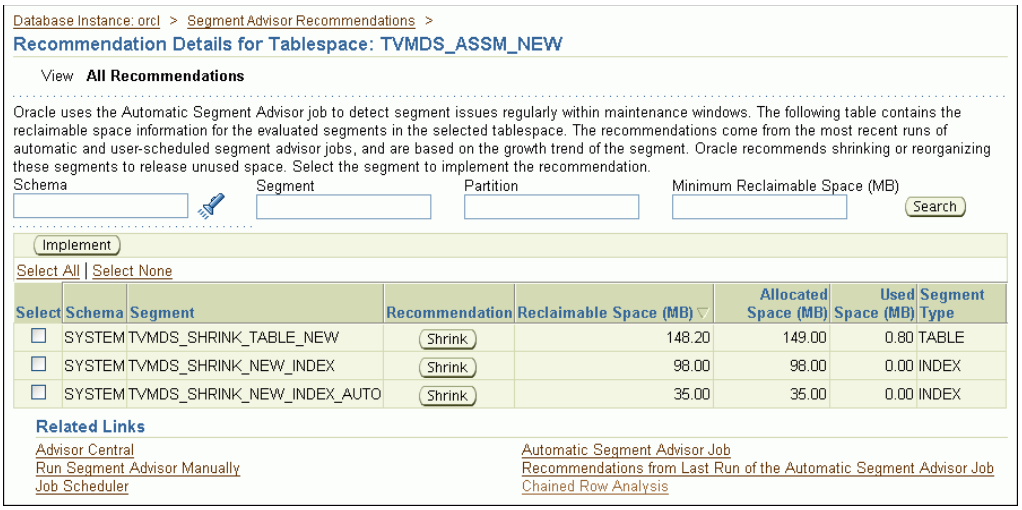
For more information about a page, at any time, click **Help**.

3. (Optional) In the View list, select **Recommendations for Last Automatic Run** to view recommendations from only the most recent automatic run of the Segment Advisor.

4. Select a tablespace, and then click **Recommendation Details**.

   The Recommendation Details for Tablespace page appears. This page lists the tablespace segments for which online segment shrink or reorganization is recommended.

Figure 6–7 Segment Advisor Recommendation Details Page



5. (Optional) Reduce the number of segments displayed in the segment list by entering search criteria in the fields **Schema**, **Segment**, **Partition**, or **Minimum Reclaimable Space**, and then clicking **Search**.
6. Examine the recommendation for each segment, and then proceed as follows:
   - If the recommendation for any segments is to reorganize, start reorganization for those segments by running subprograms from the `DBMS_REDEFINITION` PL/SQL package, using SQL*Plus. See *Oracle Database Administrator's Guide* for instructions.
   - If the recommendation for any segments is to shrink, proceed with Step 7 through Step 12 for those segments.
7. Do one of the following to select one or more segments to shrink:
   - To reclaim space in a single segment, click **Shrink** under the **Recommendation** column for that segment.
   - To shrink one or more segments, select the segments, and then click **Implement**.
8. On the Shrink Segment: Options page, accept **Compact Segments and Release Space**.

   This option returns freed space to the tablespace. If you do not want to release the freed space to the tablespace, then choose **Compact Segments**. You can rerun the shrink process later to release the freed space.

   > **Note:** Choose **Compact Segments** if you believe that you may have long-running queries currently in operation that involve the segments being shrunk. Long-running queries that were started before the shrink operation completes might attempt to read from blocks that have been reclaimed. This results in an error. Alternatively, you can schedule the shrink operation for an off-peak period in Step 10.

9. Click **Implement**.

   The Shrink Segment: Schedule page appears.

10. Note your job name, or replace it with a job name of your choosing. Under the heading Start, select **Immediately**.

    Because the shrink operation can be resource intensive, you can also select **Later** and schedule the operation for an off-peak period.

11. Click **Submit**.

    The Scheduler Jobs page appears, and shows the shrink job in the Running subpage.

12. Click **Refresh** to update the page until the job disappears from the Running subpage.

    You can then switch to the History or All subpage to view the completed status of the job.

Running the Segment Advisor Manually

You can run the Segment Advisor manually. You do so when you want to analyze objects not selected for analysis by the Automatic Segment Advisor, or when you want more up-to-date recommendations on a tablespace.

See Also:

- *Oracle Database Administrator's Guide* for more information about running the Segment Advisor

Managing the Redo Log

This section describes some of the more common redo log management tasks. It contains the following topics:

- Multiplexing the Redo Log
- Switching a Log File

See Also:

- "Step 8 - Recovery Configuration" on page 2-11 for conceptual information about redo logs
- "About Online Redo Log Files" on page 6-3
- "Viewing Online Redo Log File Information" on page 6-10

Multiplexing the Redo Log

Oracle recommends that you multiplex the redo log. Multiplexing provides better protection for data in the case of instance or media failure.

To multiplex your redo log, you must add members to each redo log group. It is not required that redo log groups be symmetrical, but Oracle recommends that your groups all have the same number of members. A database must have a minimum of two redo log groups.

Note: When you multiplex the redo log, the database must increase the amount of I/O that it performs. Depending on your configuration, this may impact overall database performance.

To multiplex the redo log:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Redo Log Groups**.

   The Redo Log Groups page appears.

   For more information about a page, at any time, click **Help**.

4. Select a group and click **Edit**.

   The Edit Redo Log Group page appears.

5. Click **Add**.

   The Add Redo Log Member page appears.

6. In the **File Name** field, enter a file name for the new redo log member.

   For example, if your existing member file name is `REDO01.log`, then you might name this member `REDO01a.log`.

7. In the **File Directory** field, enter the file directory or accept the default.

   You can create this file in the same directory, but it is recommended that you store members on separate disk drives. That way, if there is a drive failure, you still have access to one member.

8. Click **Continue**, and then click **Apply** to accept your changes.

   A confirmation message appears.

9. At the top of the page, click the **Redo Log Groups** link to return to the Redo Log Groups page.
10. Repeat Step 4 through Step 9 for every existing log group.

Switching a Log File

When a log switch occurs, the log writer (LGWR) process stops writing to the current redo log group and starts writing to the next available redo log group. You can force a log switch to make the current redo group inactive and available for redo log maintenance operations. For example, you might want to drop the current redo group, but are not able to do so until the group is inactive. You may also want to force a log switch if the current redo group needs to be archived at a specific time before the members of the group are completely filled. This option is useful in configurations with large redo log files that take a long time to fill.

To switch a log file:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Storage section, click **Redo Log Groups**.

   The Redo Log Groups page appears.

   For more information about a page, at any time, click **Help**.

4. In the Actions list, select **Switch logfile**, and then click **Go**.

A confirmation message appears. The status of the group that had been Current changes to Active, and the status of the next group in the list changes from Inactive to Current.

Managing Undo Data

Beginning with Oracle Database Release 11*g*, for a default installation, Oracle Database automatically manages the undo data. There is typically no need for DBA intervention. However, if your installation uses Oracle Flashback operations, you may need to perform some undo management tasks to ensure the success of these operations.

This section provides background information and instructions for managing undo data. It contains the following topics:

- About Undo Data
- About Managing Undo Data
- Viewing Undo Information
- Computing the Minimum Undo Tablespace Size Using the Undo Advisor
- Changing the Undo Tablespace to a Fixed Size

About Undo Data

When a transaction modifies data, Oracle Database copies the original data before modifying it. The original copy of the modified data is called **undo data**. Saving this information is necessary for the following reasons:

- To undo any uncommitted changes made to the database in the event that a rollback operation is necessary. A rollback operation can be the result of a user who wants to undo the changes of a misguided or unintentional transaction, or it can be part of a recovery operation.
- To provide read consistency, which means that each user can get a consistent view of data, even while other changes may be occurring against the data. For example, if a user issues a query at 10:00 a.m. and the query lasts for 15 minutes, then the query results should reflect the entire state of the data at 10:00 a.m., regardless of update or insert operations by other users during the query.
- To enable certain Oracle Flashback features, such as Oracle Flashback Query and Oracle Flashback Table, which enable you to view or recover data to a previous point in time.

Undo Tablespace

With automatic undo management, undo data is stored in an undo tablespace. Undo tablespaces have some additional properties beyond those of permanent tablespaces. There can be multiple undo tablespaces, but only one can be active for an Oracle instance.

When you create the database using DBCA, it creates an autoextending undo tablespace named `UNDOTBS1`, with a maximum extension size of 35 GB.

Undo Retention

Oracle Database automatically ensures that undo data that is in use by an active transaction is never overwritten until that transaction has been committed. After the transaction has been committed, the space occupied by that undo data can be reused,

or overwritten. In this case, that undo data could be overwritten if space in the undo tablespace becomes scarce.

Even after a transaction has been committed, it is useful to retain (not overwrite) its undo data, to ensure the success of Oracle Flashback features and for read consistency for long-running queries. To this end, the database maintains and automatically tunes an *undo retention period*. Committed undo data whose age is less than the undo retention period is retained for use by queries or Oracle Flashback operations.

See Also:

- *Oracle Database Concepts* for more information about read consistency
- *Oracle Database Advanced Application Developer's Guide* for more information about Oracle Flashback features

About Managing Undo Data

Although by default Oracle Database manages undo data and the undo tablespace automatically, if your installation uses Oracle Flashback features, you may need to perform some undo management tasks to ensure the success of these operations.

Oracle Flashback operations resulting in `snapshot too old` errors indicate that you must intervene to ensure that sufficient undo data is retained to support these operations.

There are two ways to better support Oracle Flashback operations:

- Set the minimum undo retention period for the autoextending tablespace to be as long as the longest expected Oracle Flashback operation.

  You do this by setting the `UNDO_RETENTION` initialization parameter. See *Oracle Database Administrator's Guide* for details.

- Change the undo tablespace to a fixed size.

  For an autoextending undo tablespace, Oracle Database always automatically tunes the undo retention period to be slightly longer than the longest-running active query. However, this autotuned retention period may be insufficient to accommodate Oracle Flashback operations.

  For an undo tablespace of a fixed size, or a tablespace with autoextending disabled, rather than tuning the undo retention period to be slightly longer than the longest-running active query, the database dynamically tunes the undo retention period for the *best possible retention*, based on system activity and the undo tablespace size. This *best possible retention* period for an undo tablespace of fixed size can be much longer than the longest-running active query, and can thus better accommodate Oracle Flashback operations.

  Another reason to change the undo tablespace to a fixed size is to prevent the tablespace from growing too large.

  If you decide to change the undo tablespace to a fixed size, you must choose a tablespace size that is sufficiently large. If you choose an undo tablespace size that is too small, the following two errors could occur:

  – DML could fail because there is not enough space to accommodate undo data for new transactions.

  – Long-running queries could fail with a `snapshot too old` error, which means that there was insufficient undo data for read consistency.

Oracle Enterprise Manager Database Control (Database Control) includes an Undo Advisor to help you determine the minimum size for the fixed size of the undo tablespace. See "Computing the Minimum Undo Tablespace Size Using the Undo Advisor" on page 6-28.

Note: If you want to configure the undo tablespace to have a fixed size, Oracle suggests that you first allow enough time after database creation to run a full workload, thus allowing the undo tablespace to grow to its minimum required size to handle the workload. Then, you can use the Undo Advisor to determine the best size to configure the undo tablespace to allow for future long-running queries and Oracle Flashback operations.

See Also:

- "About Undo Data" on page 6-25
- "Computing the Minimum Undo Tablespace Size Using the Undo Advisor" on page 6-28
- "Changing the Undo Tablespace to a Fixed Size" on page 6-30

Viewing Undo Information

You can use the Automatic Undo Management page to view the following information about your undo configuration:

- Name and current size of the undo tablespace
- Autoextend tablespace setting (Yes or No)
- Current autotuned undo retention period

To view undo information:

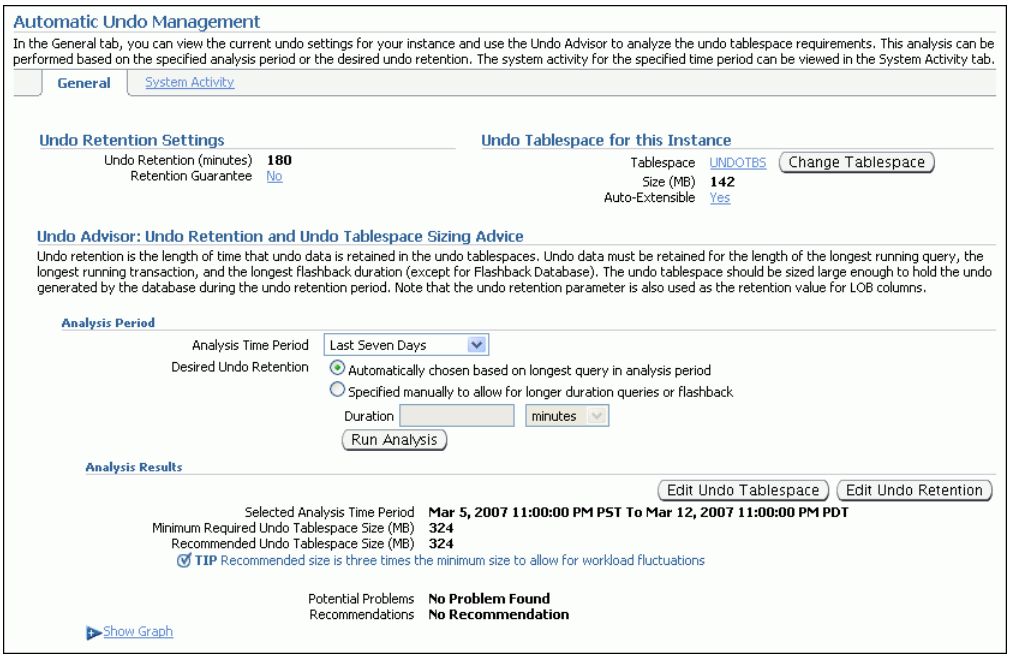
1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Database Configuration section, click **Automatic Undo Management**.

   The Automatic Undo Management page appears.

   For more information about a page, at any time, click **Help**.



See Also:

- "About Undo Data" on page 6-25

Computing the Minimum Undo Tablespace Size Using the Undo Advisor

If you decide to change the undo tablespace to a fixed size, use the Undo Advisor to help determine the minimum required size. You can also use the Undo Advisor to set the minimum undo retention period.

To compute the minimum undo tablespace size using the Undo Advisor:

1. Go to the Automatic Undo Management page.

   See "Viewing Undo Information" on page 6-27.

2. Decide whether you want to compute the minimum size of the undo tablespace based on statistics gathered over a designated time period (such as the last 7 days), or based on an undo retention period of a duration that you choose.

   The automatically gathered statistics include the duration of the longest-running query and the undo generation rate. Computing the minimum undo tablespace size based on these statistics is advisable if you do not use Oracle Flashback features or if you do not expect future long-running queries to exceed the duration of previous long-running queries.

   If you prefer to choose and enter the duration of a desired undo retention period, the duration must be based on your expectations of the duration of future long-running queries or Oracle Flashback operations.

3. If you want to compute the minimum undo tablespace size based on statistics gathered over a period of time, complete the following steps:

   a. In the Analysis Period section, in the Analysis Time Period list, select the desired analysis time period.

      If you select Customize Time Period, a page appears that enables you to enter the starting and ending date and time for the period.

b. Select **Automatically chosen based on longest query in analysis period**.

c. Click **Run Analysis**.

The minimum required undo tablespace size is displayed in the Analysis Results section.

d. (Optional) Click **Show Graph** to view a graph of the analysis results.

4. If you want to compute the minimum undo tablespace size based on a duration that you enter, complete the following steps:

a. In the Analysis Period section, select **Specified manually to allow for longer duration queries or flashback**.

b. In the **Duration** field, enter the desired duration of the undo retention period in seconds, minutes, hours or days.

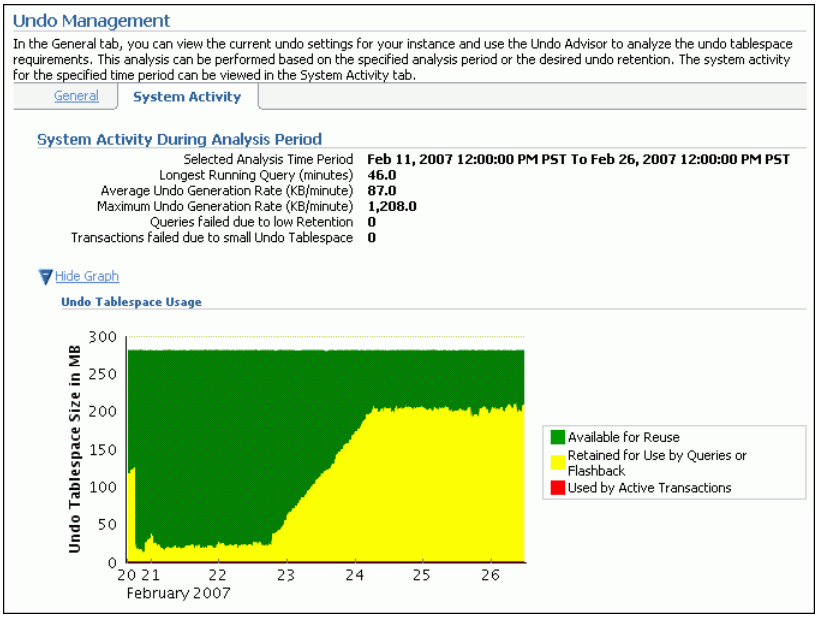
You may compute this duration as follows:

– Determine the duration of the expected longest-running query for your workload.

You can base your determination on the longest-running query recorded during a previous workload period. This information is available from the System Activity subpage of the Automatic Undo Management page. You choose the analysis period for this subpage by selecting from the Analysis Time Period list on the General subpage. You can choose a custom analysis period that corresponds to your typical workload period.

– Determine the longest duration that is expected for an Oracle Flashback operation.

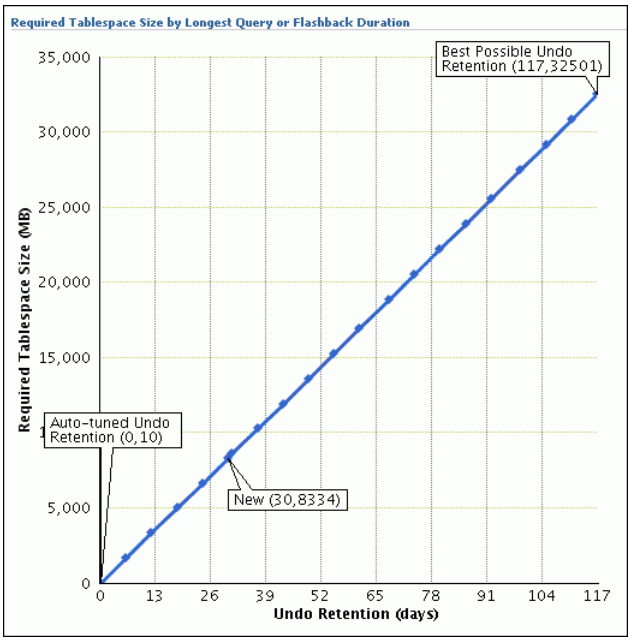
– Take the maximum of these two durations.



c. In the Analysis Period section of the General subpage, click **Run Analysis**.

The minimum required undo tablespace size is displayed in the Analysis Results section.

d. (Optional) Click **Show Graph** to view a graph of the analysis results.



5. (Optional) Click the tick-mark, or point, on the curve that corresponds to the desired undo retention period.

   The Duration field changes to match the selected undo retention period, and the Minimum Required Undo Tablespace Size field above the graph changes to reflect the matching required size.

6. (Optional) Click **Apply**.

   The minimum undo retention period is set to the value of the Duration field.

Note: Running an analysis or setting the minimum undo retention with the Undo Advisor does not change the size of the undo tablespace. You must follow the instructions in "Changing the Undo Tablespace to a Fixed Size" on page 6-30 to change the size of the undo tablespace.

See Also:

- "About Undo Data" on page 6-25
- "About Managing Undo Data" on page 6-26

Changing the Undo Tablespace to a Fixed Size

You change the undo tablespace to a fixed size to prevent the tablespace from growing too large or to better support Oracle Flashback operations.

To change the undo tablespace to a fixed size:

1. On the Automatic Undo Management page, after determining the minimum required undo tablespace size, click **Edit Undo Tablespace**.

   The Edit Tablespace page appears, displaying the properties of the undo tablespace.

2. In the Datafiles section, click **Edit**.

The Edit Datafile page appears.

3. In the **File Size** field, enter the computed minimum size for the undo tablespace.

   See "Computing the Minimum Undo Tablespace Size Using the Undo Advisor" on page 6-28.

4. In the Storage section, deselect **Automatically extend datafile when full (AUTOEXTEND)**.

5. Click **Continue**.

   The Edit Tablespace page returns.

6. Click **Apply**.

   A confirmation message appears.

See Also:

- "About Undo Data" on page 6-25
- "About Managing Undo Data" on page 6-26

Storage: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Storage OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/storage/storage.htm
```

7

Administering User Accounts and Security

This chapter describes how to create and manage user accounts. It contains the following sections:

- About User Accounts
- About User Privileges and Roles
- About Administrative Accounts and Privileges
- Administering Roles
- Administering Database User Accounts
- Setting the Database Password Policy
- Users: Oracle By Example Series

About User Accounts

For users to access your database, you must create user accounts and grant appropriate database access privileges to those accounts. A user account is identified by a user name and defines the attributes of the user, including the following:

- Authentication method
- Password (encrypted) for database authentication
- Default tablespaces for permanent and temporary data storage
- Tablespace quotas
- Account status (locked or unlocked)
- Password status (expired or not)

When you create a user account, you must not only assign a user name, a password, and default tablespaces for the account, but you must also do the following:

- Grant the appropriate system privileges, object privileges, and roles to the account.
- If the user will be creating database objects, give the user account a space usage quota on each tablespace in which the objects will be created.

Oracle recommends that you grant each user just enough privileges to perform his job, and no more. For example, a database application developer needs privileges to create and modify tables, indexes, views, and stored procedures, but does not need (and should not be granted) privileges to drop (delete) tablespaces or recover the database. You can create user accounts for database administration, and grant only a subset of administrative privileges to those accounts. In addition, you may want to create user

accounts that are used by applications only. That is, nobody logs in with these accounts; instead, applications use these accounts to connect to the database, and users log in to the applications. This avoids giving application users the ability to log in to the database directly, where they could unintentionally cause damage. See "About User Privileges and Roles" on page 7-3 for more information.

When you create a user account, you are also implicitly creating a schema for that user. A **schema** is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user. For example, `hr.employees` refers to the table named `employees` in the `hr` schema. (The `employees` table is owned by `hr`.) The terms *database object* and *schema object* are used interchangeably.

When you delete a user, you must either simultaneously delete all schema objects of that user, or you must have previously deleted the schema objects in separate operations.

Predefined User Accounts

In addition to the user accounts that you create, the database includes a number of user accounts that are automatically created upon installation.

All databases include the administrative accounts `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP`. **Administrative accounts** are highly privileged accounts, and should be used only by individuals authorized to perform administrative tasks such as starting and stopping the database, managing database memory and storage, creating and managing database users, and so on. You log in to Oracle Enterprise Manager Database Control (Database Control) with `SYS`, `SYSTEM`, or `SYSMAN`. The Management Agent of Database Control uses the `DBSNMP` account to monitor and manage the database. You assign the passwords for these accounts when you create the database with Oracle Database Configuration Assistant (DBCA). You must not delete these accounts.

All databases also include **internal accounts**, which are automatically created so that individual Oracle Database features or components can have their own schemas. An example is the `WKSYS` account, which is used by Oracle Ultra Search. All Oracle Ultra Search database objects are installed in the `WKSYS` schema. To protect these accounts from unauthorized access, they are initially locked and their passwords are expired. (A **locked account** is an account for which login is disabled.) You must not delete internal accounts, and you must not use them to log in to the database.

Your database may also include **sample schemas**, which are a set of interlinked schemas that enable Oracle documentation and Oracle instructional materials to illustrate common database tasks. These schemas also provide a way for you to experiment without endangering production data. Each sample schema has a user account associated with it. For example, the `hr` user account owns the `hr` schema, which contains a set of simple tables for a human resources application. The sample schema accounts are also initially locked and have an expired password. As the database administrator, you are responsible for unlocking these accounts and assigning passwords to these accounts.

See Also:

- *Oracle Database 2 Day + Security Guide*
- "Locking and Unlocking User Accounts" on page 7-17
- "About Administrative Accounts and Privileges" on page 7-4
- "Administering Database User Accounts" on page 7-9
- *Oracle Database Sample Schemas* for a description of the sample schemas

About User Privileges and Roles

User privileges provide a basic level of database security. They are designed to control user access to data and to limit the kinds of SQL statements that users can execute. When creating a user, you grant privileges to enable the user to connect to the database, to run queries and make updates, to create schema objects, and more. There are two main types of user privileges:

- **System privileges**—A system privilege gives a user the ability to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the system privilege `CREATE TABLE` permits a user to create tables in the schema associated with that user, and the system privilege `CREATE USER` permits a user to create database users.
- **Object privileges**—An object privilege gives a user the ability to perform a particular action on a specific schema object. Different object privileges are available for different types of schema objects. The privilege to select rows from the `EMPLOYEES` table or to delete rows from the `DEPARTMENTS` table are examples of object privileges.

Managing privileges is made easier by using **roles**, which are named groups of related privileges. You create roles, grant system and object privileges to the roles, and then grant roles to users. You can also grant roles to other roles. Unlike schema objects, roles are not contained in any schema.

Table 7–1 lists three widely used roles that are predefined in Oracle Database. You can grant these roles when you create a user or at any time thereafter.

Table 7–1 Oracle Database Predefined Roles

Role Name	Description
`CONNECT`	Enables a user to connect to the database. Grant this role to any user or application that needs database access. If you create a user using Oracle Enterprise Manager Database Control, this role is automatically granted to the user.
`RESOURCE`	Enables a user to create, modify, and delete certain types of schema objects in the schema associated with that user. Grant this role only to developers and to other users that must create schema objects. This role grants a subset of the *create object* system privileges. For example, it grants the `CREATE TABLE` system privilege, but does not grant the `CREATE VIEW` system privilege. It grants only the following privileges: `CREATE CLUSTER`, `CREATE INDEXTYPE`, `CREATE OPERATOR`, `CREATE PROCEDURE`, `CREATE SEQUENCE`, `CREATE TABLE`, `CREATE TRIGGER`, `CREATE TYPE`. In addition, this role grants the `UNLIMITED TABLESPACE` system privilege, which effectively assigns a space usage quota of `UNLIMITED` on all tablespaces in which the user creates schema objects.

Table 7–1 (Cont.) Oracle Database Predefined Roles

Role Name	Description
DBA	Enables a user to perform most administrative functions, including creating users and granting privileges; creating and granting roles; creating, modifying, and deleting schema objects in any schema; and more. It grants all system privileges, but does not include the privileges to start up or shut down the database. It is by default granted to users SYS and SYSTEM.

See Also:

- "Administering Roles" on page 7-5
- "Administering Database User Accounts" on page 7-9
- Chapter 8, "Managing Schema Objects"
- *Oracle Database 2 Day + Security Guide* for more information about privileges and roles
- *Oracle Database SQL Language Reference* for tables of system privileges, object privileges, and predefined roles

About Administrative Accounts and Privileges

Administrative accounts and privileges enable you to perform administrative functions such as managing users, managing database memory, and starting up and shutting down the database.

This section contains the following topics:

- SYS and SYSTEM Users
- SYSDBA and SYSOPER System Privileges

See Also:

- "About User Accounts" on page 7-1
- "About User Privileges and Roles" on page 7-3
- "Administering Database User Accounts" on page 7-9

SYS and SYSTEM Users

The following administrative user accounts are automatically created when you install Oracle Database. They are both created with the password that you supplied upon installation, and they are both automatically granted the DBA role.

- SYSTEM

  This account can perform all administrative functions except the following:

  – Backup and recovery

  – Database upgrade

  It is recommended that you log in with this account to perform day-to-day administrative tasks.

- SYS

  This account can perform all administrative functions. All base (underlying) tables and views for the database data dictionary are stored in the SYS schema. These

base tables and views are critical for the operation of Oracle Database. To maintain the integrity of the data dictionary, tables in the SYS schema are manipulated only by the database. They should never be modified by any user or database administrator. You must not create any tables in the SYS schema.

The SYS user is granted the SYSDBA privilege, which enables a user to perform high-level administrative tasks such as backup and recovery.

SYSDBA and SYSOPER System Privileges

SYSDBA and SYSOPER are administrative privileges required to perform high-level administrative operations such as creating, starting up, shutting down, backing up, or recovering the database. The SYSDBA system privilege is for fully empowered database administrators and the SYSOPER system privilege allows a user to perform basic operational tasks, but without the ability to look at user data.

The SYSDBA and SYSOPER system privileges allow access to a database instance even when the database is not open. Control of these privileges is therefore completely outside of the database itself. This enables an administrator who is granted one of these privileges to connect to the database instance to start the database.

You can also think of the SYSDBA and SYSOPER privileges as types of connections that enable you to perform certain database operations for which privileges cannot be granted in any other way. For example, if you have the SYSDBA privilege, then you can connect to the database using AS SYSDBA.

The SYS user is automatically granted the SYSDBA privilege upon installation. When you log in as user SYS, you must connect to the database as SYSDBA. Connecting as a SYSDBA user invokes the SYSDBA privilege. Oracle Enterprise Manager Database Control does not permit you to log in as user SYS without connecting as SYSDBA.

When you connect with SYSDBA or SYSOPER privileges, you connect with a default schema, not with the schema that is generally associated with your user name. For SYSDBA this schema is SYS; for SYSOPER the schema is PUBLIC.

Caution: When you connect as user SYS, you have unlimited privileges on data dictionary tables. Be certain that you do not modify any data dictionary tables.

See Also:

- *Oracle Database Administrator's Guide* for the operations authorized with the SYSDBA and SYSOPER privileges

Administering Roles

Roles are named groups of related system and object privileges. You create roles and then assign them to users and to other roles.

This section contains the following topics:

- Viewing Roles
- Example: Creating a Role
- Example: Modifying a Role
- Deleting a Role

See Also:

- "About User Privileges and Roles" on page 7-3
- *Oracle Database 2 Day + Security Guide* for more information about administering user security, roles, and privileges

Viewing Roles

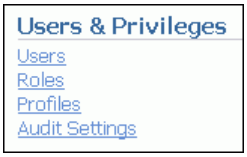
You view roles on the Roles page of Oracle Enterprise Manager Database Control (Database Control).

To view roles:

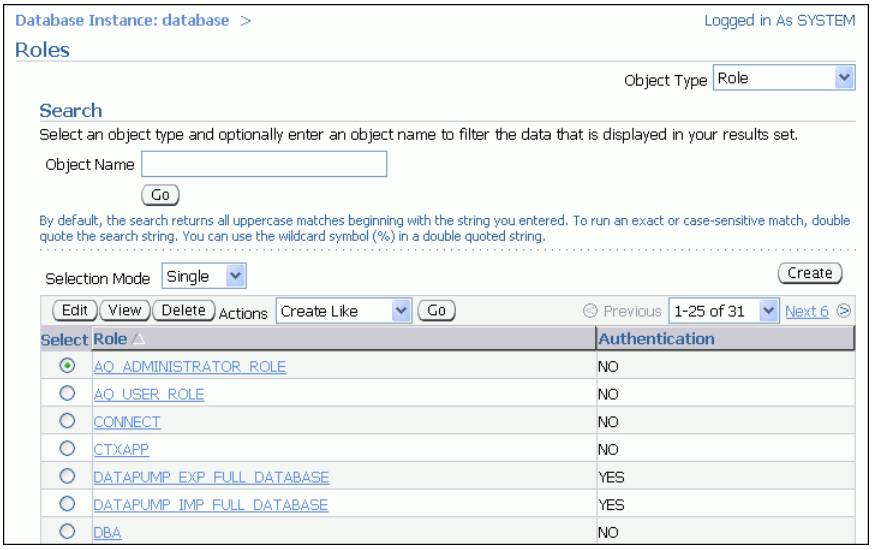
1. Go to the Database Home page, logging in with a user account that has privileges to manage roles. An example of such a user account is `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Schema** link to view the Schema subpage.
3. In the Users & Privileges section, click **Roles**.



   The Roles page appears.



4. If you want to view the details of a particular role, then in the **Select** column, select the name of the role you want to view, and then click **View**.

   If you do not see the role that you want to view, it may be on another page. In this case, do one of the following:

   - Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
   - Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

You can then select the role and click **View**.

The View Role page appears. In this page, you can see all the privileges and roles granted to the selected role.

Example: Creating a Role

Suppose you want to create a role called APPDEV for application developers. Because application developers must be able to create, modify, and delete the schema objects that their applications use, you want the APPDEV role to include the system privileges shown in Table 7–2.

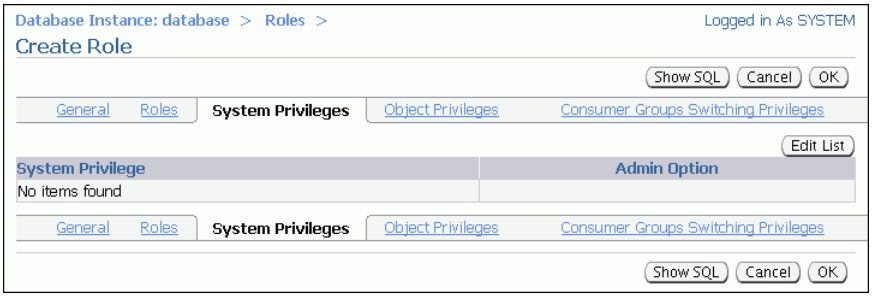
Table 7–2 System Privileges Granted to the APPDEV Role

Privilege	Description
CREATE TABLE	Enables a user to create, modify, and delete tables in his schema.
CREATE VIEW	Enables a user to create, modify, and delete views in his schema.
CREATE PROCEDURE	Enables a user to create, modify, and delete procedures in his schema.
CREATE TRIGGER	Enables a user to create, modify, and delete triggers in his schema.
CREATE SEQUENCE	Enables a user to create, modify, and delete sequences in his schema.
CREATE SYNONYM	Enables a user to create, modify, and delete synonyms in his schema.

To create the APPDEV role:

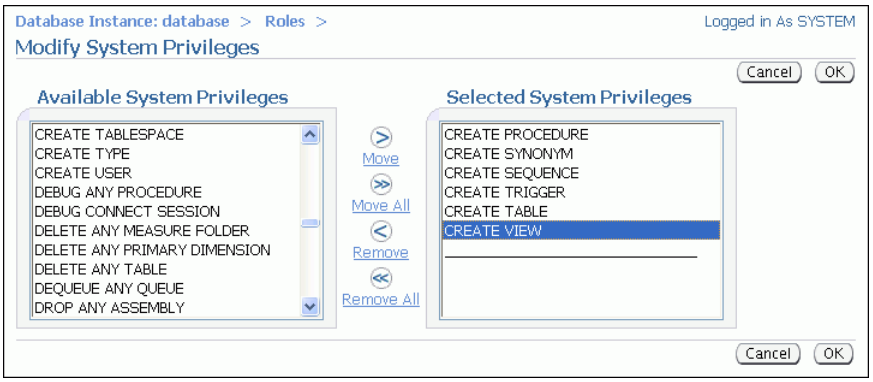
1. Go to the Roles page, as described in "Viewing Roles" on page 7-6.
2. Click **Create**.

   The Create Role page appears.
3. In the **Name** field, enter **APPDEV**.
4. Click **System Privileges** to go to the System Privileges subpage.



   The table of system privileges for this role contains no rows yet.
5. Click **Edit List**.

   The Modify System Privileges page appears.
6. In the Available System Privileges list, double-click privileges to add them to the Selected System Privileges list. The privileges to add are listed in table Table 7–2 on page 7-7.



Note: Double-clicking a privilege is a shortcut. You can also select a privilege and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges, then click **Move** after you have selected the privileges.

7. Click **OK**.

   The System Privileges subpage returns, showing the system privileges that you selected. At this point, you could click **Roles** to assign other roles to the APPDEV role, or click **Object Privileges** to assign object privileges to the APPDEV role.

8. Click **OK** to return to the Roles page.

   The APPDEV role now appears in the table of database roles.

Example: Modifying a Role

Suppose your applications make use of Oracle Streams Advanced Queuing, and you determine that developers need to be granted the roles AQ_ADMINISTRATOR_ROLE and AQ_USER_ROLE to develop and test their applications. You must edit the APPDEV role to grant it these two Advanced Queuing roles.

To modify the APPDEV role:

1. Go to the Roles page, as described in "Viewing Roles" on page 7-6.
2. In the **Select** column, click APPDEV role, and then click **Edit**.

   The Edit Role page appears.
3. Click **Roles** to navigate to the Roles subpage.
4. Click **Edit List**.

   The Modify Roles page appears.
5. In the Available Roles list, double-click the roles AQ_ADMINISTRATOR_ROLE and AQ_USER_ROLE to add them to the Selected Roles list.
6. Click **OK**.

   The Roles subpage returns, showing that the roles that you selected were granted to the APPDEV role.
7. Click **Apply** to save your changes.

An update message appears, indicating that the role APPDEV was modified successfully.

Deleting a Role

You must use caution when deleting a role, because Database Control deletes a role even if that role is currently granted to one or more users. Before deleting a role, you may want to determine if the role has any grantees. Dropping (deleting) a role automatically removes the privileges associated with that role from all users that had been granted the role.

To determine if a role has any grantees:

1. Go to the Roles page as described in "Viewing Roles" on page 7-6.
2. In the **Select** column, click the desired role.

   If you do not see the desired role, it may be on another page. In this case, do one of the following:

   - Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
   - Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

   You can then select the role.
3. In the Actions list, select **Show Grantees**, and then click **Go**.

   A report appears, listing the users that are granted the selected role.
4. Click **Cancel** to return to the Roles page.

To delete a role:

1. If you are not already there, go to the Roles page, as described in "Viewing Roles" on page 7-6.
2. In the **Select** column, click the desired role, and then click **Delete**.

   A confirmation page appears.
3. Click **Yes**.

   A confirmation message indicates that the role has been deleted successfully.

Administering Database User Accounts

This section provides instructions for creating and managing user accounts for the people and applications that use your database. It contains the following topics:

- Viewing User Accounts
- Example: Creating a User Account
- Creating a New User Account by Duplicating an Existing User Account
- Example: Granting Privileges and Roles to a User Account
- Example: Assigning a Tablespace Quota to a User Account
- Example: Modifying a User Account
- Locking and Unlocking User Accounts

- Expiring a User Password
- Example: Deleting a User Account

See Also:

- "About User Accounts" on page 7-1

Viewing User Accounts

You view user accounts on the Users page of Oracle Enterprise Manager Database Control (Database Control).

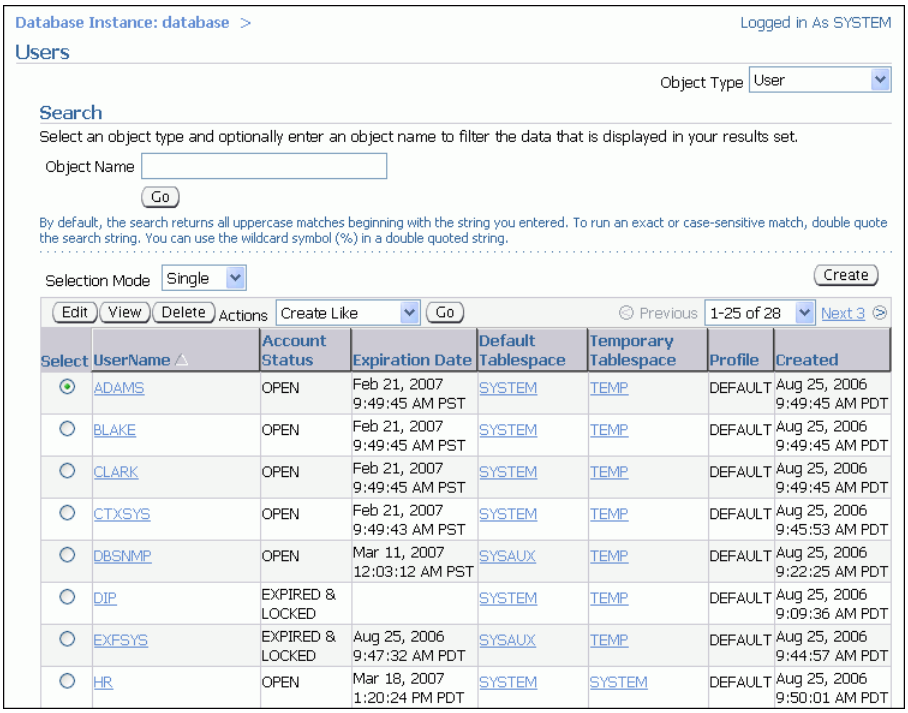
To view users:

1. Go to the Database Home page, logging in with a user account that has privileges to manage users. An example of such a user account is `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click the **Schema** link to view the Schema subpage.
3. In the Users & Privileges section of the Schema page, click **Users**.

   The Users page appears.



4. If you want to view the details of a particular user, then in the **Select** column, click the user, and then click **View**.

   If you do not see the user that you want to view, it may be on another page. In this case, do one of the following:

   - Just above the list of users, click **Next** to view the next page. Continue clicking **Next** until you see the desired user.
   - Use the Search area of the page to search for the desired user. In the **Object Name** field, enter the first few letters of the user name, and then click **Go**.

You can then select the user and click **View**.

The View User page appears, and displays all user attributes.

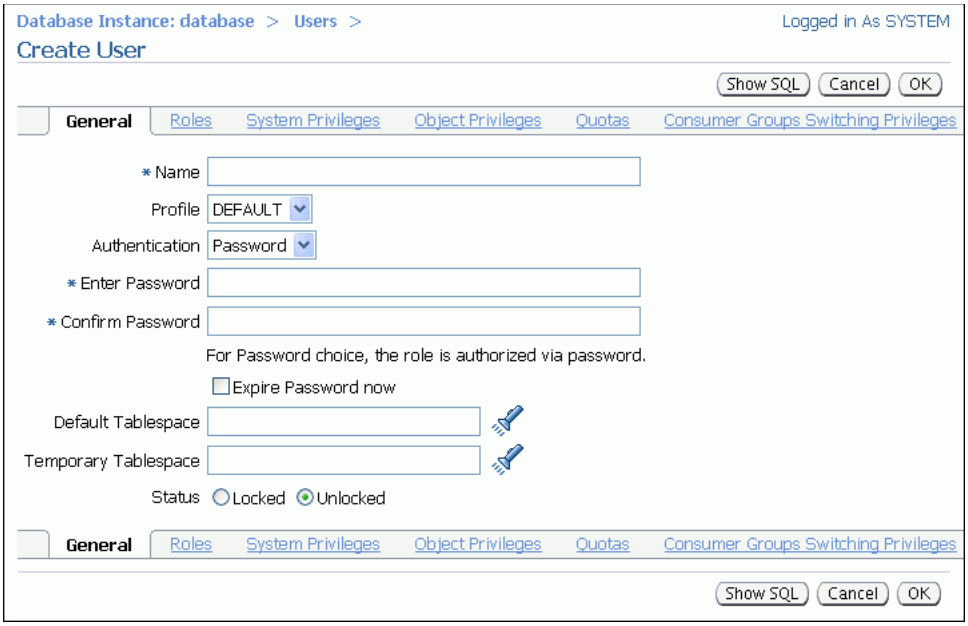
Example: Creating a User Account

Suppose you want to create a user account for a database application developer named Nick, who has requested the password "firesign007." Because Nick is a developer, you+ want to grant him the database privileges and roles that he requires to build and test his applications. You also want to give Nick a 10 megabyte (MB) quota on his default tablespace so that he can create schema objects in that tablespace.

To create the user Nick:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. On the Users page, click **Create**.

   The Create User page appears, displaying the General subpage.



3. In the **Name** field, enter **`NICK`**.
4. Accept the value `DEFAULT` in the Profile list. This assigns the default password policy to user Nick.

   See "Setting the Database Password Policy" on page 7-18.
5. Accept the default value `Password` in the Authentication list.

   For information about advanced authentication schemes, see *Oracle Database 2 Day + Security Guide*.
6. In the **Enter Password** and **Confirm Password** fields, enter a password, for example, firesign007.
7. Do not select Expire Password now. If the account status is set to expired, then the user or the database administrator must change the password before the user can log in to the database.
8. (Optional) Next to the **Default Tablespace** field, click the flashlight icon, select the **`USERS`** tablespace, and then click **Select**.

All schema objects that Nick creates will then be created in the USERS tablespace unless he specifies otherwise. If you leave the Default Tablespace field blank, Nick is assigned the default tablespace for the database, which is USERS in a newly installed database. For more information about the USERS tablespace, see "About Tablespaces" on page 6-6.

9. (Optional) Next to the **Temporary Tablespace** field, click the flashlight icon, select the **TEMP** tablespace, and then click **Select**.

   If you leave the Temporary Tablespace field blank, Nick is assigned the default temporary tablespace for the database, which is TEMP in a newly installed database. For more information about the TEMP tablespace, see "About Tablespaces" on page 6-6.

10. For the Status option, accept the default selection of **Unlocked**.

    You can later lock the user account to prevent users from logging in with it. To temporarily deny access to a user account, locking the user account is preferable to deleting it, because deleting it also deletes all schema objects owned by the user.

11. Grant roles, system privileges, and object privileges to the user, as described in "Example: Granting Privileges and Roles to a User Account" on page 7-13.

    Note: Do not click OK in Step 13 of "Example: Granting Privileges and Roles to a User Account" on page 7-13. Instead, skip that step and continue with Step 12 in this procedure.

12. Assign a 10 MB quota on the USERS tablespace, as described in "Example: Assigning a Tablespace Quota to a User Account" on page 7-15.

13. If you did not click OK while assigning the tablespace quota (previous step), click **OK** now to create the user.

See Also:

- "Creating Database Control Administrative Users" on page 3-6
- *Oracle Database 2 Day + Security Guide*.

Creating a New User Account by Duplicating an Existing User Account

If you want to create a user account that is similar in attributes to an existing user account, you can duplicate the existing user account.

To create a new user account by duplicating an existing user account:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. In the **Select** column, click the user that you want to duplicate.
3. In the Actions list, select **Create Like**, and then click **Go**.

   The Create User page appears. This page displays a new user with the same attributes as the duplicated user.

4. Enter a user name and password, modify the user attributes or privileges if desired, and then click **OK** to save the new user.

The Actions list also provides shortcuts for other actions, and provides a way to display the SQL command used to create a user.

Example: Granting Privileges and Roles to a User Account

Suppose you are creating or modifying a user account named Nick. Because Nick is a database application developer, you want to grant him the APPDEV role, which enables him to create database objects in his own schema. (You created the APPDEV role in "Example: Creating a Role" on page 7-7.) Because you want Nick to be able to create tables and views in other schemas besides his own, you want to grant him the CREATE ANY TABLE and CREATE ANY VIEW system privileges. In addition, because he is developing a human resources application, you want him to be able to view the tables in the hr sample schema and use them as examples. You therefore want to grant him the SELECT object privilege on those tables. Finally, you want Nick to be able to log in to Database Control so that he can use the graphical user interface to create and manage his database objects. You therefore want to grant him the SELECT ANY DICTIONARY system privilege. The following table summarizes the privileges and roles that you want to grant to Nick.

Grant Type	Privilege or Role Name
System privileges	CREATE ANY TABLE, CREATE ANY VIEW, and SELECT ANY DICTIONARY
Object privileges	SELECT on all tables in the hr schema
Roles	APPDEV

The following example assumes that you are already in the process of creating the user account for Nick or editing the account. This means that you have already accessed the Create User page and have entered all required fields on the General subpage (see "Example: Creating a User Account" on page 7-11), or that you have already accessed the Edit User page for Nick (see "Example: Modifying a User Account" on page 7-16). The example also assumes that you have not yet granted any privileges or roles to Nick.

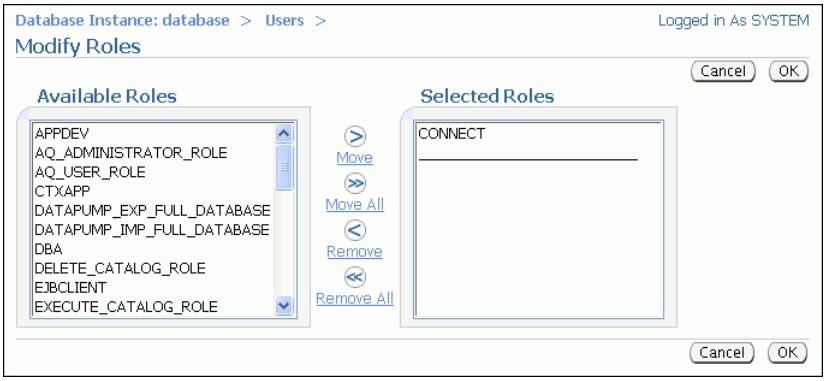
To grant privileges and roles to the user Nick:

1. Toward the top of the Create User or Edit User page, click **Roles** to display the Roles subpage.

   The Roles subpage shows that the CONNECT role is assigned to Nick. Database Control automatically assigns this role to all users that you create. (The selected Default check box indicates that the CONNECT role is a **default role** for Nick, which means that it is automatically enabled whenever Nick logs in.)

2. Click **Edit List**.

   The Modify Roles page appears.



3. In the Available Roles list, locate the **APPDEV** role, double-click it to add it to the Selected Roles list, and then click **OK**.

   The Create User or Edit User page returns, showing that both the CONNECT and APPDEV roles are granted to Nick.

   Note: Double-clicking a role is a shortcut. You can also select the role and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges, then click **Move** after you have selected the privileges.

4. Toward the top of the page, click **System Privileges** to select the System Privileges subpage.
5. Click **Edit List**.

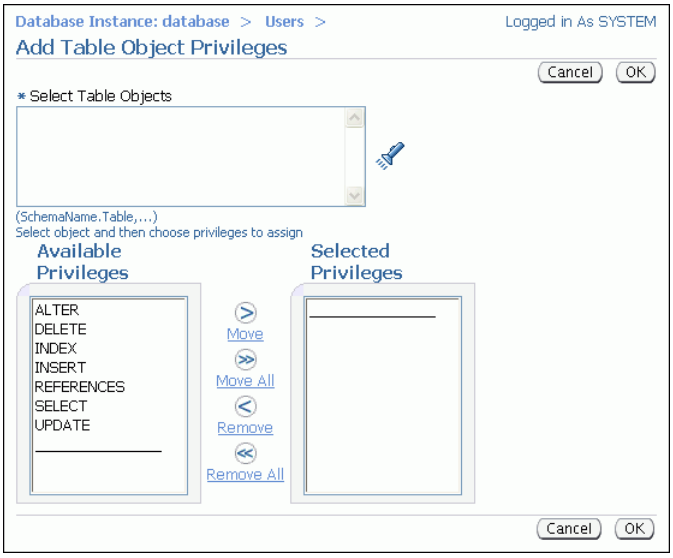
   The Modify System Privileges page appears.
6. In the Available System Privileges list, scroll to locate the CREATE ANY TABLE, CREATE ANY VIEW, and SELECT ANY DICTIONARY privileges, double-click each to add them to the Selected System Privileges list, and then click **OK**.

   The Create User or Edit User page returns, showing the newly added system privileges.

   Note: To revoke a role, double-click it in the Selected Roles list on the Modify Roles page. To revoke a system privilege, double-click it in the Selected System Privileges list on the Modify System Privileges page.

7. Toward the top of the page, click **Object Privileges** to select the Object Privileges subpage.
8. In the Select Object Type list, select **Table** and then click **Add**.

   The Add Table Object Privileges page appears.



9. Click the flashlight icon next to the Select Table Objects list.

   The Select Table Objects dialog box appears.

10. In the Schema list, select **HR**, and then click **Go**.

    All tables in the `hr` schema are displayed.

11. Click **Select All**, and then click the **Select** button.

    The Select Table Objects dialog box closes, and the names of all tables in the `hr` schema appear in the Select Table Objects field on the Add Table Object Privileges page.

12. In the Available Privileges list, double-click the **SELECT** privilege to move it to the Selected Privileges list, and then click **OK**.

    The Create User or Edit User page returns, showing that the `SELECT` object privilege for all `hr` tables is granted to user Nick.

    Note: To revoke an object privilege, select it on the Create User or Edit User page (Object Privileges subpage), and then click **Delete**.

13. Do one of the following to save the role and privilege grants:

    - If you are creating a user account, click **OK** to save the new user account.
    - If you are modifying a user account, click **Apply** to save the changes for the user account.

    See Also:

    - "About User Privileges and Roles" on page 7-3
    - *Oracle Database 2 Day + Security Guide*

Example: Assigning a Tablespace Quota to a User Account

Suppose you are creating or modifying a user account named Nick. You want to assign Nick a space usage quota of 10 MB on his default tablespace.

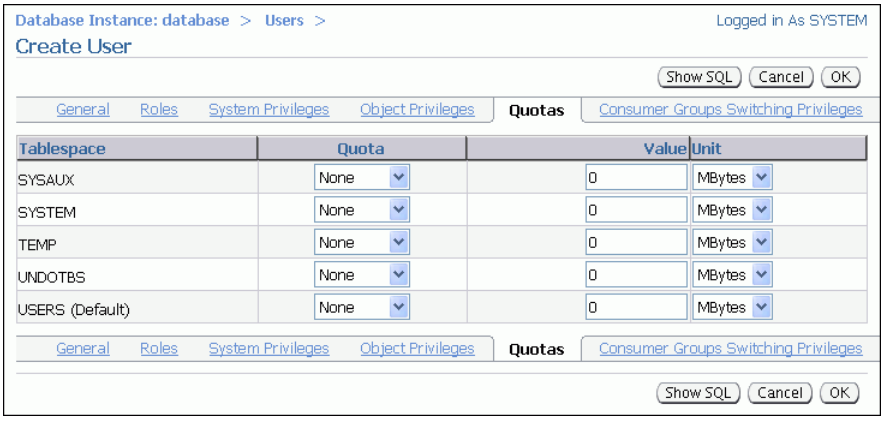
You must assign Nick a tablespace quota on his default tablespace before he can create objects in that tablespace. (This is also true for any other tablespace in which Nick wants to create objects.) After a quota is assigned to Nick for a particular tablespace, the total space used by all of his objects in that tablespace cannot exceed the quota. You can also assign a quota of `UNLIMITED`.

The following example assumes that you are already in the process of creating the user account for Nick or editing the account. This means that you have already accessed the Create User page and have entered all required fields on the General subpage (see "Example: Creating a User Account" on page 7-11), or that you have already accessed the Edit User page for Nick (see "Example: Modifying a User Account" on page 7-16). The example also assumes that Nick has not yet been assigned a quota on any tablespaces.

To assign a tablespace quota to user Nick:

1. Toward the top of the Create User or Edit User page, select the **Quotas** subpage.

   The Quotas subpage appears, showing that user Nick does not have a quota assigned on any tablespace.



2. In the **Quota** column for tablespace `USERS`, select **Value** from the list.
3. In the **Value** column for tablespace `USERS`, enter **`10`**.
4. Do one of the following to save the new quota assignment:
   - If you are creating a user account, click **OK** to save the new user account.
   - If you are modifying a user account, click **Apply** to save changes for the user account.

Example: Modifying a User Account

Suppose you want to remove the quota limitations for the user Nick on his default tablespace, `USERS`. To do so, you must modify his user account.

To modify the user Nick:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. In the **Select** column, select the user account Nick, and then click **Edit**.

   If you do not see user Nick, he may be on another page. In this case, do one of the following:

- Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
- Use the Search area of the page to search for his account. In the **Object Name** field, enter the letters **NI**, and then click **Go**.

You can then select the user account for Nick and click **Edit**.

The Edit User page appears, and displays the general attributes for Nick.

3. Toward the top of the page, select the **Quotas** subpage.
4. In the **Quota** column for tablespace `USERS`, select **Unlimited** from the list, and then click **Apply**.

   A message appears, indicating that user Nick was modified successfully.

Locking and Unlocking User Accounts

To temporarily deny access to the database for a particular user account, you can lock the user account. If the user then attempts to connect, the database displays an error message and does not allow the connection. You can unlock the user account when you want to permit database access again for that user.

To lock or unlock a user account:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. In the **Select** column, click the desired user account.

   If you do not see the desired user account, it may be on another page. In this case, use the **Next** button to view additional pages or use the Search area of the page to search for the desired user account.
3. Do one of the following:
   - To lock the account, select **Lock User** from the Actions list, and then click **Go**.
   - To unlock the account, select **Unlock User** from the Actions list, and then click **Go**.

   A confirmation message appears.
4. Click **Yes**.

Expiring a User Password

When you expire a user password, the user is prompted to change his or her password the next time that user logs in. Reasons to expire a password include the following:

- A user password becomes compromised.
- You have a security policy in place that requires users to change their passwords on a regular basis.

> **Note:** You can automate the automatic expiring of user passwords after a certain interval. See "Setting the Database Password Policy" on page 7-18.

- A user has forgotten his or her password.

In this third case, you modify the user account, assign a new temporary password, and expire the password. The user then logs in with the temporary password and is prompted to choose a new password.

To expire a user password:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. In the **Select** column, click the desired user account.

   If you do not see the desired user account, it may be on another page. In this case, do one of the following:

   - Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the desired user account.
   - Use the Search area of the page to search for the desired user account. In the **Object Name** field, enter the first few letters of the user account name, and then click **Go**.

   You can then select the user account.
3. Select **Expire Password** from the Actions list, and then click **Go**.

   A confirmation message appears.
4. Click **Yes** to complete the task.

Example: Deleting a User Account

Suppose user Nick has moved to another department. Because it is no longer necessary for him to have access to the database, you want to delete his user account.

You must use caution when deciding to deleting a user account, because this action also deletes all schema objects owned by the user. To prevent a user from logging in to the database while keeping the schema objects intact, lock the user account instead. See "Locking and Unlocking User Accounts" on page 7-17.

To delete user Nick:

1. Go to the Users page, as described in "Viewing User Accounts" on page 7-10.
2. In the **Select** column, select the user account Nick, and then click **Delete**.

   If you do not see the user account Nick, it may be on another page. In this case, do one of the following:

   - Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
   - Use the Search area of the page to search for the user account. In the **Object Name** field, enter the letters **NI**, and then click **Go**.

   You can then select the user account for Nick and click **Delete**.

   A confirmation page appears.
3. Click **Yes** to confirm the deletion of the user account.

Setting the Database Password Policy

This section provides background information and instructions for setting the password policy for all user accounts in the database. It contains the following topics:

- About Password Policies

- Modifying the Default Password Policy

See Also:

- "Administering Database User Accounts" on page 7-9
- *Oracle Database 2 Day + Security Guide*

About Password Policies

When you create a user account, a default password policy is assigned to that user account. The default password policy for a newly installed database includes these directives:

- The password for the user account expires automatically in 180 days.
- The user account is locked 7 days after password expiration.
- The user account is locked for 1 day after 10 failed login attempts.

The default password policy is assigned to user accounts through a database object called a *profile*. Each user account is assigned a profile, and the profile has a number of attributes that together describe a password policy. The database comes with a default profile (named `DEFAULT`), and unless you specify otherwise when you create a user account, the default profile is assigned to the user account.

For better database security, you may want to impose a more strict password policy. For example, you may want passwords to expire every 70 days, and you may want to lock user accounts after three failed login attempts. (A failed login attempt for a user account occurs when a user enters an incorrect password for the account.) You may also want to require that passwords be complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords. For example, you might specify that passwords must contain at least one number and one punctuation mark.

You change the password policy for every user account in the database by modifying the password-related attributes of the `DEFAULT` profile.

Note: It is possible to have different password policies for different user accounts. You accomplish this by creating multiple profiles, setting password-related attributes differently for each profile, and assigning different profiles to different user accounts. This scenario is not addressed in this section.

Modifying the Default Password Policy

You modify the default password policy for every database user account by modifying the password-related attributes of the profile named `DEFAULT`.

To modify the default password policy:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

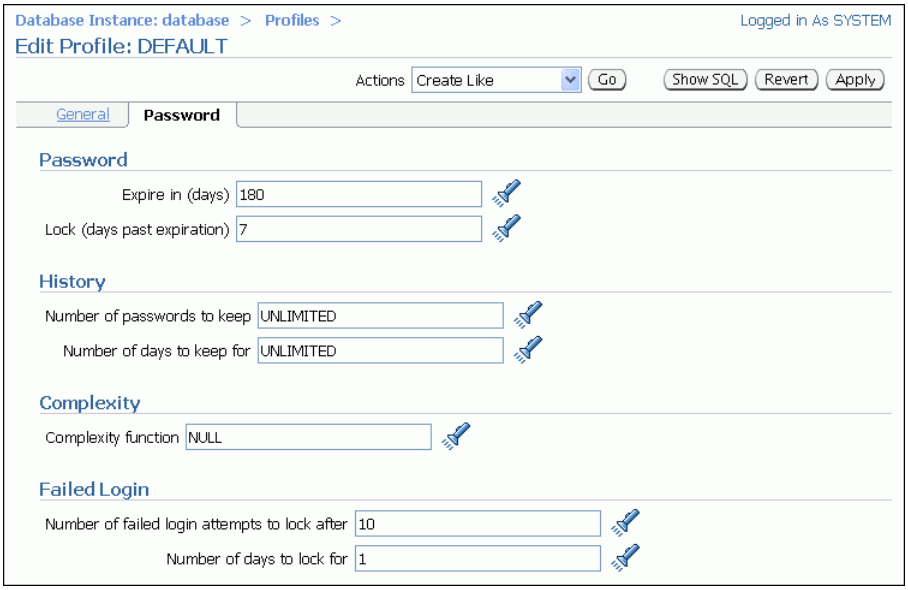
2. At the top of the page, click the **Server** link to view the Server subpage.
3. In the Security section, click **Profiles**.

   The Profiles page appears.

4. In the **Select** column, select the profile named **`DEFAULT`**, and then click **Edit**.

The Edit Profile page appears.

5. Toward the top of the page, select the **Password** subpage.



6. Change field values as required. Click the flashlight icon next to each field to view a list of choices. (Click **Help** on this page for a description of the fields.)
7. Click **Apply** to save your changes.

See Also:

- "About Password Policies" on page 7-19
- *Oracle Database 2 Day + Security Guide*

Users: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Users OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/users/users.htm
```

8

Managing Schema Objects

This chapter discusses the creation and management of schema objects. It contains the following sections:

- About Schema Objects
- About Schema Object Management Privileges
- Managing Tables
- Managing Indexes
- Managing Views
- Managing Program Code Stored in the Database
- Working with Other Schema Objects
- Schemas: Oracle By Example Series

About Schema Objects

A **schema** is a collection of database objects. A schema is owned by a database user and shares the same name as the user. **Schema objects** are logical structures created by users. Some objects, such as tables or indexes, hold data. Other objects, such as views or synonyms, consist of a definition only.

> **Note:** There is no relationship between a tablespace and a schema. Objects in the same schema can use storage in different tablespaces, and a tablespace can contain data from different schemas.

Naming Schema Objects

Every object in the database belongs to one schema and has a unique name within that schema. Multiple database objects can share the same name, if they are in different schemas. You can use the schema name to unambiguously refer to objects. For example, `hr.employees` refers to the table named `employees` in the `hr` schema. (The `employees` table is owned by `hr`.) The terms *database object* and *schema object* are used interchangeably.

When you create a database object, you must ensure that you create it in the intended schema. One way to do this is to log in to the database as the user who owns the schema and then create the object. Generally, you place all the objects that belong to a single application in the same schema.

A schema object name must abide by certain rules. In addition to being unique within a schema, a schema object name cannot be longer than 30 bytes and must begin with a

letter. If you attempt to create an object with a name that violates any of these rules, the database raises an error.

The Show SQL Button

You can create and manipulate schema objects with SQL or with Oracle Enterprise Manager Database Control (Database Control).

When creating schema objects using Database Control, you can click the **Show SQL** button to display the SQL statement that is the equivalent of the schema object properties that you specified with the graphical user interface. Database Control submits this SQL statement to create the schema object. This option shows the statement even if it is incomplete, so you must enter all specifications for the schema object to see the complete SQL statement that Database Control submits.

See Also:

- *Oracle Database SQL Language Reference* for more detailed information about schema objects, object names, and data types

About Schema Object Management Privileges

As a database administrator (DBA), you can create, modify, and delete schema objects in your own schema and in any other schema. For purposes of this discussion, a database administrator is defined as any user who is granted the `DBA` role. This includes the `SYS` and `SYSTEM` users by default.

You can enable other users to manage schema objects without necessarily granting them DBA privileges. For example, a common scenario is to enable an application developer to create, modify, and delete schema objects in his or her own schema. To do so, you grant the `RESOURCE` role to the application developer.

As described in "Access to Database Control for Nonadministrative Users" on page 3-6, you must also grant the developer access to Database Control before he or she can log in to Database Control to manage schema objects. If you do not grant access to Database Control, the developer must manage schema objects with SQL*Plus or SQL Developer.

See Also:

- "Example: Granting Privileges and Roles to a User Account" on page 7-13
- "About SQL and SQL-Based Management Tools" on page 3-11

Managing Tables

The following topics discuss database tables and how to create and modify them:

- About Tables
- Viewing Tables
- Viewing Table Data
- Example: Creating a Table
- Modifying Table Attributes
- Example: Loading Data into a Table
- Deleting a Table

About Tables

The table is the basic unit of data storage in an Oracle database. It holds all user-accessible data. Each table is made up of columns and rows. In the `employees` table, for example, there are columns called `last_name` and `employee_id`. Each row in the table represents a different employee, and contains a value for `last_name` and `employee_id`.

When you create a table, you specify the table type, and define its columns and constraints. **Constraints** are rules that help preserve data integrity.

This section contains the following topics:

- About Table Types
- About Table Column Attributes
- About Table-Level Constraints
- About Table Storage Attributes
- Other Table Creation Considerations

About Table Types

The most common type of table in an Oracle database is a *relational* table, which is structured with simple columns similar to the `employees` table. Two other table types are supported: *object* tables and *`XMLType`* tables. Any of the three table types can be defined as *permanent* or *temporary*. Temporary tables hold session-private data that exists only for the duration of a transaction or session. They are useful in applications where a results set must be held temporarily in memory, perhaps because the results set is constructed by running multiple operations.

You can build relational tables in either *heap* or *index-organized* structures. In heap structures, the rows are not stored in any particular order. In index-organized tables, the row order is determined by the values in one or more selected columns. For some applications, index-organized tables provide enhanced performance and more efficient use of disk space.

This section describes permanent, heap-organized tables. For information about other table types and when to use them, see *Oracle Database Administrator's Guide*, *Oracle Database Concepts*, and *Oracle Database Performance Tuning Guide*. For the syntax required to create and alter tables with SQL, see *Oracle Database SQL Language Reference*.

About Table Column Attributes

You define table columns to hold your data. When you create a column, you specify the following attributes:

- Data Type
- NOT NULL Column Constraint
- Default Value
- Encryption

Data Type The data type attribute defines the kind of data to be stored in the column. When you create a table, you must specify a data type for each of its columns.

Data types define the domain of values that each column can contain. For example, `DATE` columns cannot accept the value `February 29` (except for a leap year) or the values `2` or `SHOE`. Each value subsequently inserted in a column assumes the column

data type. For example, if you insert `17-JAN-2004` into a date column, then Oracle Database treats that character string as a date value after verifying that it converts to a valid date.

Table 8–1 lists some common Oracle Database built-in data types.

Table 8–1 Common Data Types

Data Type	Description
`VARCHAR2(`*`size`* `[BYTE|CHAR])`	Variable-length character string having a maximum length of *`size`* bytes or characters. A column to hold postal codes for different countries, for example, might be restricted to 12 bytes by defining it as `VARCHAR2(12)`. You can use the `CHAR` qualifier, for example `VARCHAR2(10 CHAR)`, to indicate the maximum length in characters, without regard for the number of bytes required. This is especially useful for languages that use characters with double-byte and triple-byte lengths. The `BYTE` and `CHAR` qualifiers override the setting of the `NLS_LENGTH_SEMANTICS` parameter, which has a default of bytes. The maximum size is 4000 bytes or characters. The minimum is 1 byte or 1 character. You must specify *`size`* for `VARCHAR2`. See *Oracle Database Globalization Support Guide* for more information.
`NUMBER (`*`p`*`,`*`s`*`)`	Number having precision *`p`* and scale *`s`*. Precision sets the maximum number of digits in the number, and scale defines how many of the digits are to the right of the decimal point. For example, a field to hold monetary values might be defined as `NUMBER(12,2)`, providing 10 digits for the primary unit of currency (dollars, pounds, marks, and so on) and two digits for the secondary unit (cents, pennies, pfennigs, and so on). The precision *`p`* can range from 1 to 38. The scale *`s`* can range from -84 to 127.
`DATE`	A composite value that includes both a date and time component. For each `DATE` value, the database stores the following information: century, year, month, day, hour, minute, and second. When entering a date into a table column of type `DATE`, you must use the format specified by the `NLS_DATE_FORMAT` initialization parameter. The `NLS_TERRITORY` initialization parameter determines the default value of the `NLS_DATE_FORMAT` parameter. For example, in the United States, the `NLS_DATE_FORMAT` parameter defaults to '`DD-MON-RR`'. You must therefore enter a date in the format '`11-JAN-06`'. Because this format does not include a time component, the time defaults to 12:00:00 a.m. (midnight). You can also use the `TO_DATE` function, which converts a character string to a date, to include a time component or to enter a date in another format. The valid date range is from January 1, 4712 BC to December 31, 9999 AD.
`CLOB`	A character large object (CLOB) containing single-byte or multibyte characters. Both fixed-width and variable-width character sets are supported, both using the database character set. The maximum size is (4 gigabytes - 1) * (database block size). For example, for a block size of 32K, the maximum CLOB size is 128 terabytes.

See Also:

- *Oracle Database SQL Language Reference* for a complete list of Oracle Database built-in data types, and for details about date formatting and the `TO_DATE` function
- *Oracle Database Globalization Support Guide* for more information about the `NLS_DATE_FORMAT` and `NLS_TERRITORY` parameters

NOT NULL Column Constraint Constraints determine valid values for the column. In Oracle Enterprise Manager Database Control (Database Control), the only constraint you can define at the column level on the Create Table page is the `NOT NULL`

constraint, which requires that a value be included in the column whenever a row is inserted or updated. Unlike other constraints described in "About Table-Level Constraints" on page 8-5, which can be defined as part of the column definition or part of the table definition, the `NOT NULL` constraint must be defined as part of the column definition.

Use a `NOT NULL` constraint when data must be supplied for a column for the integrity of the database. For example, if all employees must belong to a specific department, then the column that contains the department identifier must be defined with a `NOT NULL` constraint. On the other hand, do not define a column as `NOT NULL` if the data can be unknown or may not exist when rows are added or changed. An example of a column for which you must not use a `NOT NULL` constraint is the second, optional line in a mailing address.

The database automatically adds a `NOT NULL` constraint to the column or columns included in the primary key of a table.

Default Value This value is automatically stored in the column whenever a new row is inserted without a value being provided for the column. You can specify a default value as a literal or as an expression. However, there are limitations on how you construct the expression. See *Oracle Database SQL Language Reference* for details.

Encryption You can enable automatic encryption for column data. See the discussion of transparent data encryption in *Oracle Database 2 Day + Security Guide* for more information.

About Table-Level Constraints

In an Oracle database, you can apply rules to preserve the integrity of your data. For example, in a table that contains employee data, the employee name column cannot accept `NULL` as a value. Similarly, in this table, you cannot have two employees with the same ID.

Oracle Database enables you to apply data integrity rules called **constraints**, both at the table level and at the column level. Any SQL statement that attempts to insert or update a row that violates a constraint results in an error, and the statement is rolled back. Likewise, any attempt to apply a new constraint to a populated table also results in an error if any existing rows violate the new constraint.

The types of constraints that you can apply at the table level are as follows:

- **Primary Key**—Requires that a column (or combination of columns) be the unique identifier of the row. A primary key column does not allow `NULL` values.
- **Unique Key**—Requires that no two rows can have duplicate values in a specified column or combination of columns. The set of columns is considered to be a unique key.
- **Check**—Requires that a column (or combination of columns) satisfy a condition for every row in the table. A check constraint must be a Boolean expression. It is evaluated each time that a row is inserted or updated. An example of a check constraint is: `SALARY > 0`.
- **Foreign Key**—Requires that for a particular column (or combination of columns), all column values in the child table exist in the parent table. The table that includes the foreign key is called the dependent or **child** table. The table that is referenced by the foreign key is called the **parent** table. An example of a foreign key constraint is where the department column of the employees table must contain a department ID that exists in the parent department table.

Constraints can be created and, in most cases, modified with different statuses. The options include enabled or disabled, which determine if the constraint is checked when rows are added or modified, and deferred or immediate, which cause constraint validation to occur at the end of a transaction or at the end of a statement, respectively.

See Also:

- *Oracle Database Concepts* for more information about constraints

About Table Storage Attributes

You can specify a number of storage attributes for a table. For example, you can specify the initial size of the table on disk. For more information about setting storage attributes for a table, see *Oracle Database Administrator's Guide* and *Oracle Database SQL Language Reference*.

Other Table Creation Considerations

This section describes some additional considerations for creating tables. It contains the following topics:

- User-Defined Types and Large Objects (LOBs)
- Partitioned Tables and Indexes
- Compressed Tables

User-Defined Types and Large Objects (LOBs) Your new table can include one or more columns defined with user-defined types. **User-defined types** enable a single column in a single row to contain multiple values. The multiple values can be represented as arrays, nested tables, or objects, where an object type represents a real-world entity such as a purchase order. (Retrieving a purchase order–type column value could return a *record* that contains purchase order number, customer number, amount, and so on.) User-defined types are created with the `CREATE TYPE` statement and are described in detail in *Oracle Database SQL Language Reference*.

Large object (LOB) columns are used to contain unstructured data (such as text or streaming video), and can hold terabytes of information. In Oracle Database 11*g*, you can use SecureFiles, the next generation LOB data type, which provide high performance, easier manageability, and full backward compatibility with existing LOB interfaces. SecureFiles also offer advanced features such as intelligent data compression, deduplication and transparent encryption. The LOB implementation available in Oracle Database 10*g* Release 2 and prior releases is still supported for backward-compatibility reasons and is now referred to as BasicFiles. If you add a LOB column to a table, you can specify whether it should be created as a SecureFile or a BasicFile. If you do not specify the storage type, the LOB is created as a BasicFile to ensure backward compatibility.

When creating a table that contains one or more LOB columns, select the **LOB** column, then click **Advanced Attributes** on the General subpage of the Create Table page to specify the storage type (BasicFile or SecureFile) and the storage options for the LOB column. To specify the same storage type and storage options for all LOB columns in a table, click **Set Default LOB Attributes**.

Partitioned Tables and Indexes You can *partition* tables and indexes. Partitioning helps to support very large tables and indexes by enabling you to divide the tables and indexes into smaller and more manageable pieces called **partitions**. SQL queries and DML statements do not need to be modified to access partitioned tables and indexes. Partitioning is transparent to the application.

After partitions are defined, certain operations become more efficient. For example, for some queries, the database can generate query results by accessing only a subset of partitions, rather than the entire table. This technique (called **partition pruning**) can provide order-of-magnitude gains in improved performance. In addition, data management operations can take place at the partition level, rather than on the entire table. This results in reduced times for operations such as data loads; index creation and rebuilding; and backup and recovery.

Each partition can be stored in its own tablespace, independent of other partitions. Because different tablespaces can be on different disks, this provides a table structure that can be better tuned for availability and performance. Storing partitions in different tablespaces on separate disks can also optimize available storage usage, because frequently accessed data can be placed on high-performance disks, and infrequently retrieved data can be placed on less expensive storage.

Partitioning is useful for many types of applications that manage large volumes of data. Online transaction processing (OLTP) systems often benefit from improvements in manageability and availability, while data warehousing systems benefit from increased performance and manageability.

Compressed Tables Table Compression is suitable for both OLTP applications and data warehousing applications. Compressed tables require less disk storage and result in improved query performance due to reduced I/O and buffer cache requirements. Compression is transparent to applications and incurs minimal overhead during bulk loading or regular DML operations such as INSERT, UPDATE or DELETE. You can configure table compression on the Storage subpage of the Create Table page.

See Also:

- *Oracle Database Administrator's Guide* for design and management considerations for different table types
- *Oracle Database Concepts* and *Oracle Database VLDB and Partitioning Guide* for more information about partitioned tables and indexes
- *Oracle Database Concepts* for more information about LOBs, SecureFiles and BasicFiles
- "Example: Creating a Table" on page 8-9

Viewing Tables

You can use Database Control to list all the tables in a specified schema, and to view the definitions of individual tables.

To view tables:

1. Go to the Database Home page, logging in as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

4. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.

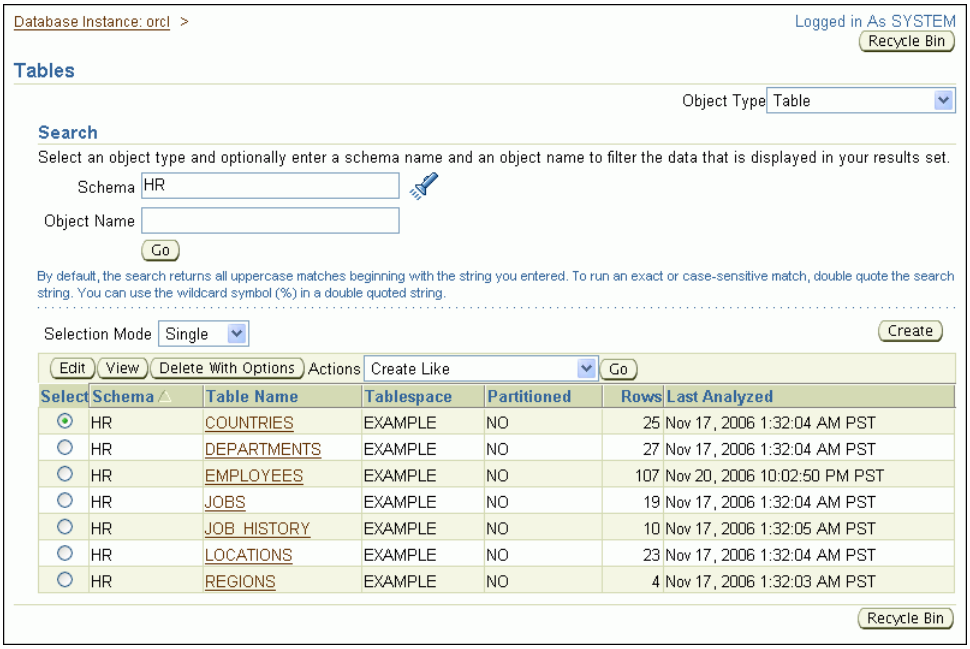
   Examples of schema names include `SYS` and `hr`.

5. Leave the Object Name field blank to search for and display all tables in the schema. Alternatively, enter a table name or partial table name to limit the search.

   If you enter a search string in the **Object Name** field, all tables that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), all tables that have the search string anywhere in the table name are displayed.

6. Click **Go**.

   The tables in the specified schema are displayed.



7. To view the definition of a particular table, select the table and then click **View**. Alternatively, click the table name.

   The View Table page appears.

See Also:

- "About Tables" on page 8-3

Viewing Table Data

Besides viewing table names and table definitions, you can view the data stored in the table, and the SQL statement used to display the data. You can also change the SQL statement to alter the results set.

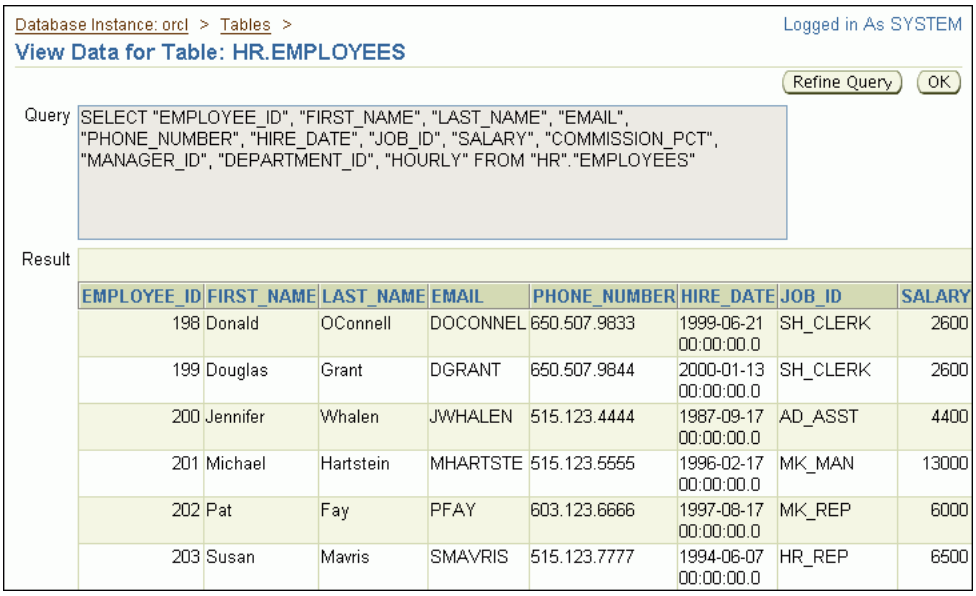
To view table data:

1. Search for a table as described in "Viewing Tables" on page 8-7. For example, search for the tables in the `hr` schema.
2. Select the table that contains the data that you want to view.

   For example, select **`employees`**.

3. In the Actions list, select **View Data**, and then click **Go**.

   The View Data for Table page appears.



The **Query** field displays the SQL query that was run to view the data for the table. The Result section shows the data in the table. You may have to use the horizontal scroll bar at the bottom of the page to view all columns.

4. (Optional) Click a column name to sort the data by that column.
5. (Optional) Click **Refine Query** to change the query and redisplay the data.

   The Refine Query for Table page appears. This page enables you to select the columns to display. It also enables you to specify a `WHERE` clause for the SQL `SELECT` statement to limit the results.

You can also write and submit your own SQL `SELECT` statement to see the contents of a table. You can run SQL statements by starting a SQL Worksheet session in Database Control. To do so, click **SQL Worksheet** in the Related Links section of the Database Home page.

A detailed description of the `SELECT` statement is in *Oracle Database SQL Language Reference*.

See Also:

- "About Tables" on page 8-3

Example: Creating a Table

You can use Database Control to create a table. Before you create and populate a table, you can estimate its size to ensure that you have sufficient space to hold its data.

In the following example, you create a table called `purchase_orders` in the `nick` schema that you created in Chapter 7, "Administering User Accounts and Security". The table has the following columns:

Column Name	Data Type	Size	Not NULL
`PO_NUMBER`	`NUMBER`		Yes
`PO_DESCRIPTION`	`VARCHAR2`	200	No
`PO_DATE`	`DATE`		Yes

Column Name	Data Type	Size	Not NULL
PO_VENDOR	NUMBER		Yes

To create the PURCHASE_ORDERS table in the NICK schema:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

4. Click **Create**.

   The Create Table: Table Organization page appears.

5. Select **Standard, Heap Organized**, and then click **Continue**.

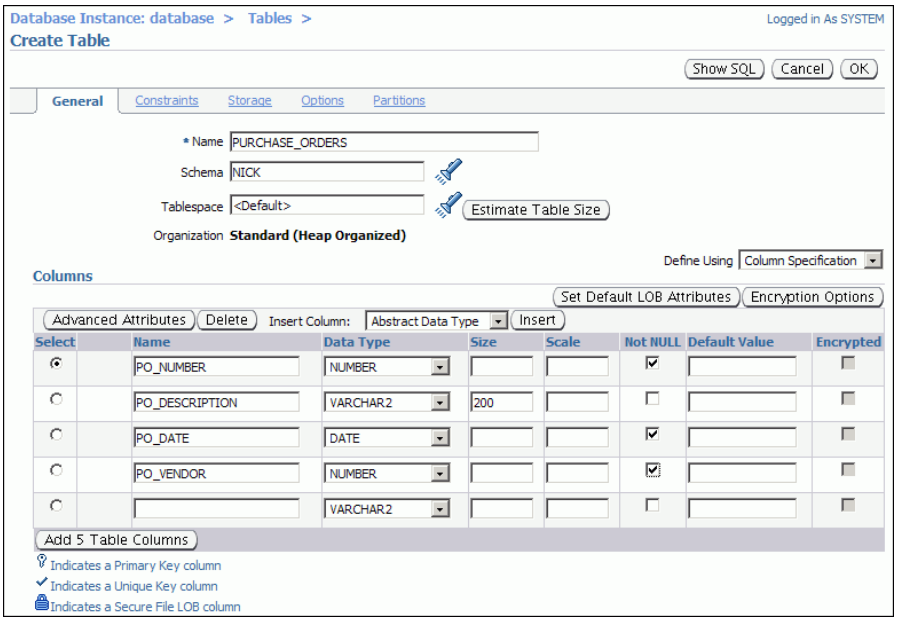
   The Create Table page appears.

6. In the **Name** field, enter **`purchase_orders`** as the table name, and in the **Schema** field, enter **`nick`**.

   Because a default tablespace was specified when creating the user `nick` in the section "Example: Creating a User Account" on page 7-11, accept the default tablespace setting for the table.

7. In the Columns section, enter column information for the `purchase_orders` table as specified in the table in the introduction to this topic. For example, for the first column in the `purchase_orders` table, enter the name **`PO_NUMBER`** and the data type **`NUMBER`**, and select the **Not NULL** check box.

   For all `purchase_orders` columns, you can leave Scale and Default Value blank.



Note: If you want to create the table with partitions, you can do so during this step by clicking **Partitions**, at the top of the page.

8. (Optional) Obtain an estimate of the table size by completing the following steps:

   a. Click **Estimate Table Size**.

      The Estimate Table Size page appears.

   b. In the **Projected Row Count** field, enter **`400000`** (four hundred thousand), and then click **Estimate Table Size**.

      The estimated results are calculated and displayed.

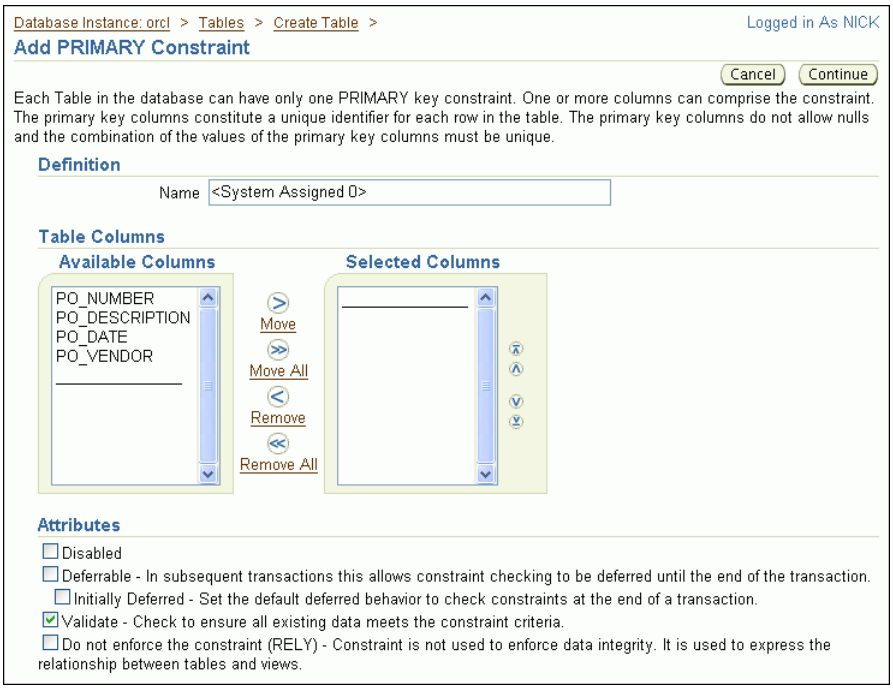
   c. Click **OK** to return to the Create Table page.

   The estimate of the table size can help you determine what values to use when specifying the storage parameters for the table.

9. Click **Constraints** to view the Constraints subpage, where you can designate a primary key for the table.

10. In the Constraints list, select **`PRIMARY`**, then click **Add**.

    The Add PRIMARY Constraint page appears.



11. In the Available Columns list, select **`PO_NUMBER`**, and then click **Move**.

> **Note:** You can also double-click **`PO_NUMBER`**.

    The `po_number` column moves to the Selected Columns list.

12. Click **Continue** to return to the Constraints subpage of the Create Table page.

13. Click **OK**.

    The Tables page returns, showing a confirmation message and listing the new table in the tables list. The `purchase_orders` table is now created with `po_number` as its primary key.

See Also:

- "About Tables" on page 8-3

Modifying Table Attributes

You can use Database Control to add and delete table columns and to manage table constraints. This section contains the following topics:

- Example: Adding Table Columns
- Example: Deleting a Table Column
- Example: Adding a New Table Constraint
- Example: Modifying an Existing Table Constraint
- Example: Deleting a Table Constraint

See Also:

- "About Tables" on page 8-3

Example: Adding Table Columns

In this example, you add columns to the `purchase_orders` table that you created previously in "Example: Creating a Table" on page 8-9. The two new columns are named `po_date_received` and `po_requestor_name`.

To add columns to the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

4. In the **Schema** field, enter **`nick`**, and then click **Go**.

   All tables owned by user `nick` are displayed.

5. Select the **`PURCHASE_ORDERS`** table, and then click **Edit**.

   The Edit Table page appears.

6. In the Columns section, in the first available row, enter the following information about the new `po_date_received` column:

Field Name	Value
Name	PO_DATE_RECEIVED
Data Type	DATE

   You can leave Size, Scale, Not NULL, and Default Value blank.

7. In the next available row, enter the following information about the new `po_requestor_name` column:

Field Name	Value
Name	PO_REQUESTOR_NAME

Field Name	Value
Data Type	VARCHAR2
Size	40

You can leave Scale, Not NULL, and Default Value blank.

8. Click **Apply**.

   An update message appears indicating that the table has been modified successfully.

See Also:

- "About Tables" on page 8-3

Example: Deleting a Table Column

In this example, you delete the `po_requestor_name` column that you added to the `purchase_orders` table in "Example: Adding Table Columns" on page 8-12.

To delete the PO_REQUESTOR_NAME column:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

4. In the **Schema** field, enter **`nick`** and then click **Go**.

   All tables owned by user `nick` are displayed.

5. Select the **`PURCHASE_ORDERS`** table, and then click **Edit**.

   The Edit Table page appears.

6. In the Columns section, select the **`PO_REQUESTOR_NAME`** column, and then click **Delete**.

   The row that contained the information for the deleted column becomes blank.

7. Click **Apply**.

   An update message appears indicating that the table has been modified successfully.

See Also:

- "About Tables" on page 8-3

Example: Adding a New Table Constraint

In this example, you add a table constraint to the `purchase_orders` table that you created in "Example: Creating a Table" on page 8-9. Suppose you want to enforce the rule that the `po_date_received` value must be either the same day as, or later than, the value of `po_date`. To do this, you add a check constraint.

Note: You can also add constraints during table creation, as shown in "Example: Creating a Table" on page 8-9. In that example, you added a primary key constraint.

To add a table constraint to the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

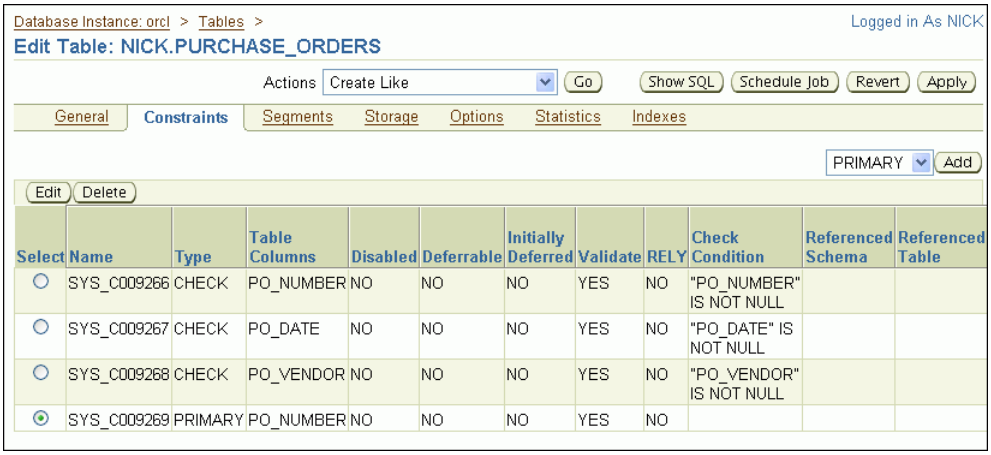
4. In the **Schema** field, enter **`nick`** and then click **Go**.

   All tables owned by user `nick` are displayed.

5. Select the **`PURCHASE_ORDERS`** table, and then click **Edit**.

   The Edit Table page appears.

6. Click **Constraints** to display the Constraints subpage.



7. In the list adjacent to the **Add** button, select **`CHECK`**, and then click **Add**.

   The Add CHECK Constraint page appears.

8. In the **Name** field, enter **`po_check_rcvd_date`**, overwriting the system-assigned default name.
9. In the **Check Condition** field, enter the following:

   ```
   po_date_received >= po_date
   ```

   This expression indicates that `po_date_received` must be greater than or equal to `po_date`. For date columns, this is equivalent to stating that `po_date_received` must be on the same day as, or later than, `po_date`.

10. Click **Continue**

    The new constraint appears on the Constraints subpage.

11. Click **Apply**.

    A confirmation message appears.

See Also:

- "About Tables" on page 8-3
- "About Table-Level Constraints" on page 8-5

Example: Modifying an Existing Table Constraint

There are a few ways in which you can modify a table constraint. You can change the status of an existing table constraint, for example, from an enabled state to a disabled state. In this example, you disable the check constraint that you created for the `purchase_orders` table in "Example: Adding a New Table Constraint" on page 8-13.

To disable a constraint for the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.

   The Tables page appears.

4. In the **Schema** field, enter **`nick`** and then click **Go**.

   All tables owned by user `nick` are displayed.

5. Select the **`purchase_orders`** table, and then click **Edit**.

   The Edit Table page appears.

6. Click **Constraints** to display the Constraints subpage.
7. Select the constraint named **`PO_CHECK_RCVD_DATE`**, and then click **Edit**.

   The Edit CHECK Constraint page appears.

8. In the Attributes section, select **Disabled**, and then click **Continue**.
9. Click **Apply**.

   A confirmation message appears. The Disabled column shows that the check constraint has been disabled.

See Also:

- "About Tables" on page 8-3
- "About Table-Level Constraints" on page 8-5

Example: Deleting a Table Constraint

You can delete constraints from a table with Database Control. Deleting a table constraint may cause the deletion of other constraints. For example, if you delete the primary key constraint from a table (the parent table) that is referenced in a foreign key constraint in another table (the child table), the foreign key constraint in the child table is also deleted through a cascading delete mechanism.

In this example, you delete the check constraint that you created for the `purchase_orders` table in "Example: Adding a New Table Constraint" on page 8-13.

To delete a constraint from the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**

   The Tables page appears.
4. In the **Schema** field, enter `NICK` and then click **Go**.

   All tables owned by user `NICK` are displayed.
5. Select the **`PURCHASE_ORDERS`** table, and then click **Edit**.

   The Edit Table page appears.
6. Click **Constraints** to display the Constraints subpage.
7. Select the constraint named **`PO_CHECK_RCVD_DATE`**, and then click **Delete**.

   The check constraint is removed from the list.
8. Click **Apply**.

   A confirmation message appears.

See Also:

- *Oracle Database Concepts* for more information about the cascading delete mechanism
- "About Tables" on page 8-3
- "About Table-Level Constraints" on page 8-5

Example: Loading Data into a Table

You can use Database Control to load data into a table. You can load data from a source file that is on your local computer—the one where your browser is running—or from a source file that is on the database host computer—the computer on which the Oracle instance is running. Because Database Control invokes the Oracle SQL*Loader utility to load the data, the format of the data in the source file can be of any format that is supported by SQL*Loader. In this example, you use a comma-delimited text file as the source file. In SQL*Loader terminology, the source file is referred to as the *data file*.

SQL*Loader also uses a control file to control the loading of data from the data file. The **control file** is a text file that contains statements written in the SQL*Loader command language. These statements specify where to find the data, how to parse and interpret the data, where to insert the data, and more. Database Control contains a Load Data *wizard* that takes you through the steps of preparing and running a data load job with SQL*Loader. (A **wizard** is an online, guided workflow.) The Load Data wizard can automatically create the SQL*Loader control file for you.

Note: The SQL*Loader control file is unrelated to the database control files described in "About Control Files" on page 6-3.

In this example, you load data into the `PURCHASE_ORDERS` table that you created in "Example: Creating a Table" on page 8-9. For simplicity, this example loads only three rows.

To prepare for this example, you must create a text file named `load.dat` on the file system of the database host computer or on the file system of your local computer. The contents of the file should be as follows:

```
1, Office Equipment, 25-MAY-2006, 1201, 13-JUN-2006
2, Computer System, 18-JUN-2006, 1201, 27-JUN-2006
3, Travel Expense, 26-JUN-2006, 1340, 11-JUL-2006
```

> **Note:** This example assumes that the columns in the PURCHASE_ORDERS table are the following: PO_NUMBER, PO_DESCRIPTION, PO_DATE, PO_VENDOR, and PO_DATE_RECEIVED. If your PURCHASE_ORDERS table does not have all these columns (or has additional columns), modify the data in the text file accordingly.

To load data into the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user SYSTEM.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Data Movement**.

   The Data Movement subpage appears.

3. In the Move Row Data section, click **Load Data from User Files**.

   The Load Data: Generate or Use Existing Control File page appears.

4. Select **Automatically Generate Control File**, and enter host computer credentials (user name and password) for the database host computer.

5. Click **Continue**.

   The first page of the Load Data wizard appears. Its title is Load Data: Data Files.

6. Follow the steps in the wizard, clicking **Next** to proceed to each new step.

   For information about using any of the wizard pages, click **Help** on that page. At the conclusion of the wizard, you submit a job that runs SQL*Loader. A job status page is then displayed. If necessary, refresh the status page until you see a succeeded (or failed) status.

7. If the job succeeded, confirm that the data was successfully loaded by doing one of the following:

   - View the table data.

     See "Viewing Table Data" on page 8-8.

   - Examine the SQL*Loader log file, which is written to the host computer directory that you designated for the SQL*Loader data file.

   > **Note:** If the job succeeds, it means only that Database Control was able to run the SQL*Loader utility. It does not necessarily mean that SQL*Loader ran without errors. For this reason, you must confirm that the data loaded successfully.

8. If the job failed, examine the SQL*Loader log file, correct any errors, and try again.

See Also:

- *Oracle Database Utilities* for more information about SQL*Loader
- "About Tables" on page 8-3

Deleting a Table

If you no longer need a table, you can delete it using Database Control. When you delete a table, the database deletes the data and dependent objects of the table (such as indexes), and removes the table from the data dictionary.

When you delete a table from a locally managed tablespace that is not the `SYSTEM` tablespace, the database does not immediately reclaim the space associated with the table. Instead, it places the table and any dependent objects in the *recycle bin*. You can then restore the table, its data, and its dependent objects from the recycle bin if necessary. You can view the contents of the recycle bin by clicking **Recycle Bin** on the Tables page. Note that users can see only tables that they own in the recycle bin. See *Oracle Database Administrator's Guide* for more information about the recycle bin, including how to view, purge, and recover tables for which you are not the owner.

To delete a table:

1. Search for the table that you want to delete, as explained in "Viewing Tables" on page 8-7.
2. Select the table, and then click **Delete With Options**.

   The Delete With Options page appears.
3. Select **Delete the table definition, all its data, and dependent objects (DROP)**.
4. Select **Delete all referential integrity constraints (CASCADE CONTRAINTS)**.
5. Click **Yes**.

   The Tables page returns and displays a confirmation message.

See Also:

- "About Tables" on page 8-3

Managing Indexes

The following topics describe how to create and manage indexes:

- About Indexes
- Viewing Indexes
- Example: Creating an Index
- Example: Deleting an Index

About Indexes

Indexes are optional schema objects that are associated with tables. You create indexes on tables to improve query performance. Just as the index in a guide helps you to quickly locate specific information, an Oracle Database index provides quick access to table data.

You can create as many indexes on a table as you need. You create each index on one or more columns of a table. For example, in a purchase orders table, if you create an index on the vendor number column, you can then sequentially access the rows of the table in vendor number order, without having to actually sort the rows. Additionally, you can directly access all purchase orders issued to a particular vendor without having to scan the entire table.

After an index is created, it is automatically maintained and used by the database. Changes to the data or structure of a table, such as adding new rows, updating rows, or deleting rows, are automatically incorporated into all relevant indexes. This is transparent to the user.

Some indexes are created implicitly through constraints that are placed on a table. For example, the database automatically creates an index on the columns of a primary key constraint or unique key constraint.

The following topics provide more background information about indexes:

- Indexes and Performance
- Index Attributes

See Also:

- "Viewing Indexes" on page 8-20
- "Example: Creating an Index" on page 8-21
- "Example: Deleting an Index" on page 8-22

Indexes and Performance

Indexes generally improve the performance of queries and DML statements that operate on a single, existing row or a small number of existing rows. However, too many indexes can increase the processing overhead for statements that add, modify, or delete rows.

To determine if you can improve application performance with more indexes, you can run the SQL Access Advisor in Oracle Enterprise Manager Database Control (Database Control). See "Running the SQL Access Advisor" on page 10-26.

Before you add additional indexes, examine the performance of your database for queries and DML. You can then compare performance after the new indexes are added.

See Also:

- "About Indexes" on page 8-18
- *Oracle Database 2 Day + Performance Tuning Guide* for information about using the SQL Performance Analyzer to analyze the SQL performance impact of any type of schema or system changes

Index Attributes

Indexes can be created in a number of ways, using various combinations of index attributes. The primary index attributes are the following:

- Standard (B-tree) and Bitmap
- Ascending and Descending
- Column and Functional
- Single-Column and Concatenated
- Nonpartitioned and Partitioned

Standard (B-tree) and Bitmap A standard, B-tree index contains an entry for each value in the index key along with a disk address of the row where the value is stored. A B-tree index is the default and most common type of index in an Oracle database.

A bitmap index uses strings of bits to encapsulate values and potential row addresses. It is more compact than a B-tree index and can perform some types of retrieval more efficiently. For general use, however, a bitmap index requires more overhead during row operations on the table and should be used primarily for data warehouse environments, as described in *Oracle Database Data Warehousing Guide*.

Ascending and Descending The default search through an index is from lowest to highest value, where character data is sorted by ASCII values, numeric data from smallest to largest number, and date from the earliest to the latest value. This default search method is performed in indexes created as ascending indexes. You can cause index searches to reverse the search order by creating the related index with the descending option.

Column and Functional Typically, an index entry is based on the value or values found in the column or columns of a table. This is a column index. Alternatively, you can create a function-based index in which the indexed value is derived from the table data. For example, to find character data that can be in various combinations of upper and lower case letters, you can use a function-based index based on the `UPPER()` function to look for the values as if they were all in uppercase characters.

Single-Column and Concatenated You can create an index on just one column, which is called a **single-column index**, or on multiple columns, which is called a **concatenated index**. Concatenated indexes are useful when all the index columns are likely to be included in the `WHERE` clause of frequently executed SQL statements.

For concatenated indexes, you must define the columns used in the index carefully so that the column with the fewest duplicate values is named first, the column with next fewest duplicate values is named second, and so on. Columns with many duplicate values or many rows with `NULL` values should not be included or should be the last-named columns in the index definition.

Nonpartitioned and Partitioned As with tables, you can partition an index. In most situations, it is useful to partition an index when the associated table is partitioned, and to partition the index using the same partitioning scheme as the table. (For example, if the table is range-partitioned by sales date, you create an index on sales date and partition the index using the same ranges as the table partitions.) This is known as a **local** partitioned index. However, you do not need to partition an index using the same partitioning scheme as its table. You can also create a nonpartitioned, or **global**, index on a partitioned table.

See Also:

- *Oracle Database Concepts* for design and management considerations of different index types
- *Oracle Database SQL Language Reference* for the syntax to create indexes
- *Oracle Database VLDB and Partitioning Guide* for more information about partitioned tables and indexes

Viewing Indexes

You use the Indexes page of Database Control to view the indexes in your database.

To view indexes:

1. Go to the Database Home page, logging in as user `SYSTEM`.

See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Indexes**.

   The Indexes page appears.
4. In the Search By list, do one of the following:
   - Select **Index Name** to search for indexes by name.

     Every index has a system-assigned or user-assigned name.
   - Select **Table Name** to search for indexes that belong to a particular table.
5. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.
6. Do one of the following:
   - If you are searching by index name, leave the Object Name field blank to search for and display all indexes in the schema. Alternatively, enter an index name or partial index name as a search string.

     If you enter a search string in the **Object Name** field, all indexes that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), all indexes that have the search string anywhere in the index name are displayed.
   - If you are searching by table name, enter a table name or partial table name in the **Object Name** field.

     If you enter a partial table name as a search string, indexes are displayed for all tables that have names that start with the search string. If you precede the search string with an asterisk (*), indexes are displayed for all tables that have the search string anywhere in the table name.
7. Click **Go**.

   The indexes in the specified schema are displayed.
8. To view the definition of a particular index, select the index and then click **View**. Alternatively, double-click the index name.

   The View Index page appears. This page includes basic information about the index, including its status, the table and column or columns on which it is built, the space used by the index, the options used in its definition, and index statistics.

See Also:

- "About Indexes" on page 8-18

Example: Creating an Index

When you create an index, you specify one or more table columns to be indexed and the type of index that you want to create.

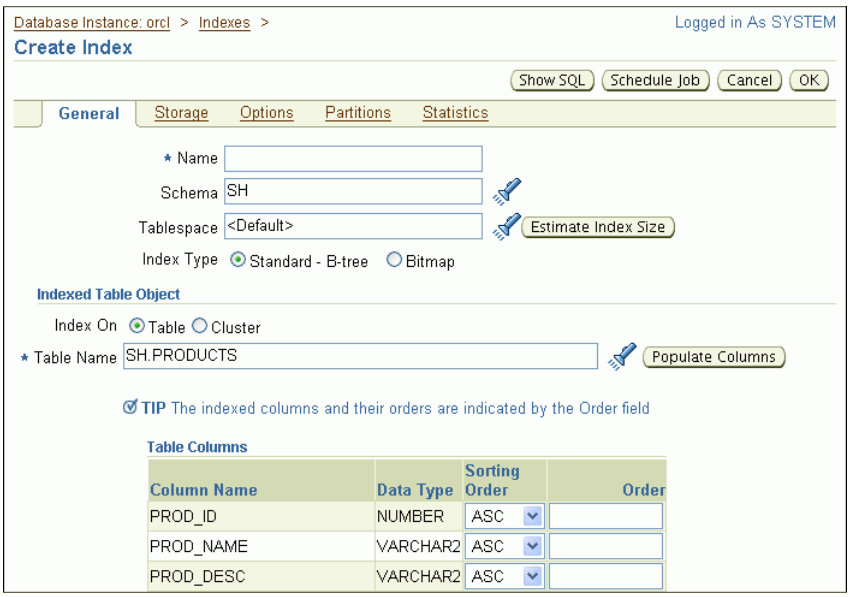
In this example, you create a standard B-tree index on the `SUPPLIER_ID` column in the `SH.PRODUCTS` table. (The `SH` schema is part of the sample schemas.)

To create a supplier index on the SH.PRODUCTS table:

1. View the tables in the `SH` schema, by following the instructions in the section "Viewing Tables" on page 8-7.

2. Select the **PRODUCTS** table.
3. In the Actions list, select **Create Index**, and then click **Go**.

   The Create Index page appears.



4. Enter the following information:
   - In the **Name** field, enter **PRODUCTS_SUPPLIER_IDX**.
   - For the **Tablespace** field, accept the default value.
   - For Index Type, select **Standard - B-tree**.
   - In the Table Columns list, select the `SUPPLIER_ID` column by entering **1** in the **Order** column.

     If your index were to consist of multiple columns (a concatenated index), you would enter `2` in the next column to include, and so on. These numbers indicate the order in which the columns are to be concatenated, from left to right, or from most significant in the sort order to least significant.
   - For **Sorting Order**, accept the default selection of **ASC** (ascending).
5. Click **OK** to create the index.

   The Indexes page returns and displays a confirmation message. The new index is listed in the table of indexes.

See Also:

- "About Indexes" on page 8-18

Example: Deleting an Index

If you no longer need an index, you can delete it using Database Control.

In this example, you delete the `PRODUCTS_SUPPLIER_IDX` index that you created previously on the `SH.PRODUCTS` table in "Example: Creating an Index" on page 8-21.

Note: You cannot delete an index that is currently used to enforce a constraint. You must disable or delete the constraint and then, if the index is not deleted as a result of that action, delete the index.

To delete the supplier index on the SH.PRODUCTS table:

1. Go to the Database Home page, logging in as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Indexes**.

   The Indexes page appears.

4. In the Search By list, select **Table Name**.
5. In the **Schema** field, enter **`SH`**.
6. In the **Object Name** field, enter **`PROD`**.

   You can enter only the first few letters of the table name.

7. Click **Go**.

   All indexes on the `PRODUCTS` table are displayed.

8. Select the **`PRODUCTS_SUPPLIER_IDX`** index, and then click **Delete**.

   A confirmation page appears.

9. Click **Yes** to delete the index.

   The Indexes page returns and displays a confirmation message.

See Also:

- "About Indexes" on page 8-18

Managing Views

The following topics describe how to create and manage views:

- About Views
- Displaying Views
- Example: Creating a View
- Example: Deleting a View

About Views

Views are customized presentations of data in one or more tables or other views. You can think of them as stored queries. Views do not actually contain data, but instead derive their data from the tables upon which they are based. These tables are referred to as the **base tables** of the view.

Similar to tables, views can be queried, updated, inserted into, and deleted from, with some restrictions. All operations performed on a view actually affect the base tables of the view. Views can provide an additional level of security by restricting access to a predetermined set of rows and columns of a table. They can also hide data complexity and store complex queries.

Many important views are in the SYS schema. There are two types: *static data dictionary views* and *dynamic performance views*. Complete descriptions of the views in the SYS schema are in *Oracle Database Reference*.

Static Data Dictionary Views

The data dictionary views are called **static views** because they change infrequently, only when a change is made to the data dictionary. Examples of data dictionary changes include creating a new table or granting a privilege to a user.

Many data dictionary tables have three corresponding views:

- A DBA_ view displays all relevant information in the entire database. DBA_ views are intended only for administrators.

  An example of a DBA_ view is DBA_TABLESPACES, which contains one row for each tablespace in the database.

- An ALL_ view displays all the information accessible to the current user, including information from the schema of the current user, and information from objects in other schemas, if the current user has access to those objects through privileges or roles.

  An example of an ALL_ view is ALL_TABLES, which contains one row for every table for which the user has object privileges.

- A USER_ view displays all the information from the schema of the current user. No special privileges are required to query these views.

  An example of a USER_ view is USER_TABLES, which contains one row for every table owned by the user.

The columns in the DBA_, ALL_, and USER_ views are usually nearly identical.

Dynamic Performance Views

Dynamic performance views monitor ongoing database activity. They are available only to administrators. The names of dynamic performance views start with the characters V$. For this reason, these views are often referred to as V$ views.

An example of a V$ view is V$SGA, which returns the current sizes of various System Global Area (SGA) memory components.

See Also:

- "Displaying Views" on page 8-24
- "Example: Creating a View" on page 8-25
- "Example: Deleting a View" on page 8-26

Displaying Views

You can use Oracle Enterprise Manager Database Control (Database Control) to list the views in a specified schema. You can also display the view definitions.

To display views:

1. Go to the Database Home page, logging in as user SYSTEM.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Views**.

The Views page appears.

4. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.

   Examples of schema names include `SYS` and `hr`.

5. Leave the Object Name field blank to search for and display all views in the schema. Alternatively, enter a view name or partial view name to limit the search.

   If you enter a search string in the **Object Name** field, all views that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), all views that have the search string anywhere in the view name are displayed.

6. Click **Go**.

   The views in the specified schema are displayed.

7. To view the definition of a particular view, select the view and then click **View**. Alternatively, double-click the view name.

   The View page appears.

See Also:

- "About Views" on page 8-23

Example: Creating a View

In this example, you create a view named `king_view`, which uses the `hr.employees` table as its base table. (The `hr` schema is part of the sample schemas.) This view filters the table data so that only employees who report directly to the manager King, whose employee ID is 100, are returned in queries. In an application scenario, this view adds an additional level of security to the `hr.employees` table while providing a suitable presentation of relevant information for manager King.

To create the KING_VIEW view on the HR.EMPLOYEES table:

1. Go to the Database Home page, logging in as user `hr` or as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.

3. In the Database Objects section, click **Views**.

   The Views page appears.

4. Click **Create**.

   The Create View page appears.

5. Enter the following information:

   - In the **Name** field, enter **`king_view`**.
   - In the **Schema** field, enter **`hr`**.
   - In the **Query Text** field, enter the following SQL statement:

     ```
     SELECT * FROM hr.employees
     WHERE manager_id = 100
     ```

6. Click **OK**.

The Views page returns and displays a confirmation message. The new view appears in the list of views.

To test the new KING_VIEW view:

1. On the Views page, select **king_view** and then select **View Data** from the Actions list.
2. Click **Go**.

   The View Data for View page appears. The data selected by the view appears in the Result section.
3. (Optional) You can also test the view by submitting the following SQL statement in SQL*Plus or SQL Developer:

```
SELECT * FROM king_view
```

See Also:

- "About Views" on page 8-23

Example: Deleting a View

If you no longer need a view, you can delete it using Database Control.

In this example, you delete the `hr.king_view` view that you created previously in "Example: Creating a View" on page 8-25.

To delete the HR.KING_VIEW view:

1. Go to the Database Home page, logging in as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Views**.

   The Views page appears.
4. In the **Schema** field, enter **hr**.
5. In the **Object Name** field, enter **king**.

   You can enter just the first few letters of the view name.
6. Click **Go**.

   `KING_VIEW` is displayed in the list of views.
7. Select **KING_VIEW**, and then click **Delete**.

   A Confirmation page appears.
8. Click **Yes** to delete the view.

   The Views page returns and displays a confirmation message.

See Also:

- "About Views" on page 8-23

Managing Program Code Stored in the Database

This section describes your responsibilities as a database administrator (DBA) with respect to program code that is stored in the database. It contains the following topics:

- About Program Code Stored in the Database
- Validating (Compiling) Invalid Schema Objects

About Program Code Stored in the Database

Oracle Database offers the ability to store program code in the database. Developers write program code in PL/SQL or Java, and store the code in schema objects. You, as the DBA, can use Oracle Enterprise Manager Database Control (Database Control) to manage program code objects such as:

- PL/SQL packages, procedures, functions, and triggers
- Java source code (Java sources) and compiled Java classes

The actions that you can perform include creating, compiling, creating synonyms for, granting privileges on, and showing dependencies for these code objects. You access administration pages for these objects by clicking links in the Programs section of the Schema subpage.

Note that creating and managing program code objects is primarily the responsibility of application developers. However, as a DBA you might have to assist in managing these objects. Your most frequent task with respect to program code objects might be to revalidate (compile) them, because they can become invalidated if the schema objects on which they depend change or are deleted.

Note: Other types of schema objects besides program code objects can become invalid. For example, if you delete a table, any views that reference that table become invalid.

See Also:

- *Oracle Database 2 Day Developer's Guide* for more information about PL/SQL procedures, functions, packages, and triggers
- *Oracle Database 2 Day + Java Developer's Guide* for more information about Java sources and Java classes
- *Oracle Database PL/SQL Language Reference* to learn about PL/SQL code
- *Oracle Database Administrator's Guide* for more information about object invalidation

Validating (Compiling) Invalid Schema Objects

As a database administrator (DBA), you may be asked to revalidate schema objects that have become invalid. Schema objects (such as triggers, procedures, or views) are invalidated when changes are made to objects on which they depend. For example, if a PL/SQL procedure contains a query on a table and the table structure is modified, the PL/SQL procedure becomes invalid. You revalidate schema objects by compiling them.

> **Note:** It is not always possible to revalidate a schema object that stores program code by compiling it. You may have to take remedial actions first. For example, if a view becomes invalid because a table that it references is deleted, compiling the view produces an error message that indicates that the table does not exist. You cannot validate the view until you re-create the table or retrieve it from the recycle bin.

Database Control notifies you when schema objects become invalid by displaying an alert in the Alerts section of the Database Home page.

There are two ways to display schema objects that require validation: by following an alert on the Database Home page, or by viewing the appropriate object page (Views page, Procedures page, and so on) and searching for the objects.

To validate schema objects starting from an alert:

1. Go to the Database Home page, logging in as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

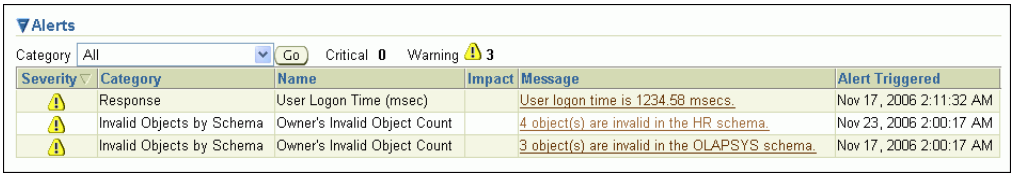
2. In the Alerts section, search for alerts with the following message:

   n `object(s) are invalid in the` *schema_name* `schema.`

   An example of such a message is the following:

   ```
   4 object(s) are invalid in the HR schema.
   ```

   There is a separate message for each schema that contains invalid objects.

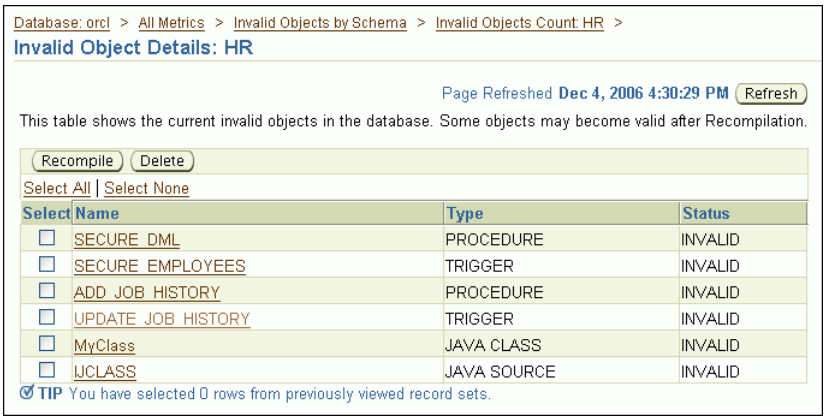


3. In the **Message** column, click an invalid object message.

   The Owner's Invalid Object Count page appears.

4. Under Related Links, click **Invalid Objects Details**.

   The Invalid Object Details page appears, showing a list of invalid objects.



5. Select one or more objects, and then click **Recompile**.

   A confirmation message appears, and the newly validated objects are removed from the list.

To validate a schema object starting from an object page:

1. Go to the Database Home page, logging in as user `SYSTEM`.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click the link for the object type that you want to validate.

   For example, if you want to validate a view, click **Views**.

4. On the object page (for example, the Views page), enter a schema name and, optionally, an object name or partial object name, and then click **Go**.

   The schema objects are displayed.

5. Select the schema object that you want to validate.
6. In the Actions list, select **Compile**, and then click **Go**.

   A confirmation message appears.

Working with Other Schema Objects

You can manage other schema objects with Oracle Enterprise Manager Database Control (Database Control), including the following:

- Sequences

  A **sequence** is a database object that generates unique integers. Each time that you query the sequence, it increments its current value by a designated amount and returns the resulting integer. Sequences can be simultaneously queried by multiple users, and each user receives a unique value. For this reason, using a sequence to provide the value for a primary key in a table is an easy way to guarantee that the key value is unique, regardless of the number of users inserting data into the table.

- Synonyms

  A **synonym** is an alias for any schema object, such as a table or view. Synonyms provide an easy way to hide the underlying database structure from an application or a user. Synonyms can be private or public. A public synonym does not need to be qualified with a schema name, whereas a private synonym does, if the user referencing the private synonym is not the synonym owner. For example, consider the following query, issued by a user who has been granted the `SELECT` object privilege on the `hr.employees` table:

  ```
  SELECT   employee_id, salary
  FROM     hr.employees
  ORDER BY salary
  ```

  Now suppose you create a public synonym named `personnel` as an alias for the `hr.employees` table, and you grant the `SELECT` privilege on the `hr.employees` table to `PUBLIC` (all database users). With the public synonym in place, any user can issue the following simpler query:

  ```
  SELECT   employee_id, salary
  FROM     personnel
  ```

```
ORDER BY salary
```

The user who created this query did not need to know the name of the schema that contains the personnel data.

> **Note:** If a user owns a table named `personnel`, that table is used in the query. If no such table exists, the database then resolves the public synonym and uses the `hr.employees` table.

An additional benefit of synonyms is that you can use the same synonym in a development database as in the production database, even if the schema names are different. This technique enables application code to run unmodified in both environments. For example, the preceding query would run without errors in a development database that had the `employees` table in the `dev1` schema, if the `personnel` synonym is defined in the development database to point to the `dev1` schema.

Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary. To reference a synonym in a query, you must have privileges on the object to which it points. Synonyms themselves cannot be secured. If you grant object privileges on a synonym to a user, you are granting privileges on the object to which the synonym points.

- Database links

  A **database link** is a schema object that points to another Oracle database. You use a database link to query or update objects in a remote database. Database links are used in distributed database environments, which are described in *Oracle Database Administrator's Guide*.

  See Also:

  - *Oracle Database 2 Day Developer's Guide* for more information about sequences
  - *Oracle Database SQL Language Reference*

Schemas: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Schemas OBE, in your browser, enter the following URL:

`http://www.oracle.com/technology/obe/11gr1_2day_dba/schema/schema.htm`

9

Performing Backup and Recovery

This chapter introduces you to Oracle Database backup and recovery with Oracle Enterprise Manager Database Control. After reading this chapter, you should be familiar with the basic concepts of Oracle Database backup and recovery operations, know how to implement a disk-based backup strategy, and perform simple repairs to database files.

This chapter contains the following sections:

- Overview of Database Backup and Recovery
- Database Backup and Recovery Concepts
- Configuring Your Database for Basic Backup and Recovery
- Backing Up Your Database
- Displaying Backup Reports
- Managing Backups
- Performing Oracle Advised Recovery
- Performing User-Directed Recovery
- Backup and Recovery: Oracle By Example Series

See Also:

- *Oracle Database Backup and Recovery User's Guide* to learn how to use the command-line interface for Recovery Manager
- *Oracle Database Backup and Recovery Reference* for the syntax and semantics of Recovery Manager commands

Overview of Database Backup and Recovery

The focus in Oracle Database backup and recovery is on the physical backup of database files, which permits you to reconstruct your database. The files protected by the backup and recovery facilities built into Oracle Enterprise Manager Database Control (Database Control) include datafiles, control files, server parameter files, and archived redo logs. With these files you can reconstruct your database. The backup mechanisms work at the physical level to protect against file damage, such as the accidental deletion of a datafile or the failure of a disk drive.

Oracle Recovery Manager (RMAN), a command-line and Enterprise Manager-based tool, is the method preferred by Oracle for efficiently backing up and recovering your Oracle database. RMAN is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes

performance and space consumption during backup with file multiplexing and backup set compression, and integrates with leading tape and storage media products.

Logical backups, such as exporting database objects such as tables or tablespaces, are a useful supplement to physical backups, but cannot protect your whole database. An effective backup strategy must be based on physical backups.

The Oracle Database flashback features provide a range of physical and logical data recovery tools as efficient, easy-to-use alternatives to physical and logical backups. The flashback features enable you to reverse the effects of unwanted database changes without restoring datafiles from backup or performing media recovery.

This section introduces the following flashback features:

- Oracle Flashback Table, which enables you to revert a table to its contents at a time in the recent past
- Oracle Flashback Drop, which enables you to retrieve deleted (dropped) database tables
- Oracle Flashback Database, which enables you to revert the entire database to a past point in time

The first two features operate at the logical level, whereas the last feature operates at the physical level. None of the preceding features requires advance preparation such as creating logical exports to allow for retrieval of your lost data. You can use all of the features while your database is available. *Oracle Database Backup and Recovery User's Guide* discusses the flashback features of Oracle Database at greater length.

Note: Oracle Flashback Database does not recover missing datafiles.

The Oracle Enterprise Manager physical backup and recovery features are built on the Recovery Manager (RMAN) command-line client. Database Control makes available many of the RMAN features, and provides wizards and automatic strategies to simplify and further automate RMAN-based backup and recovery.

Database Backup and Recovery Concepts

To **back up your database** means to make copies of your datafiles, control file, and archived redo logs (if your database runs in `ARCHIVELOG` mode). **Restoring a database** means copying the physical files that make up the database from a backup medium, typically disk or tape, to their original or to new locations. **Database recovery** is the process of updating database files restored from a backup with the changes made to the database after the backup, typically using redo log files.

Consistent and Inconsistent Backups

A backup is either consistent or inconsistent. To make a consistent backup, your database must have been shut down cleanly and remain closed for the duration of the backup. All committed changes in the redo log are written to the datafiles, so the datafiles are in a transaction-consistent state. When you restore your datafiles from a consistent backup, you can open the database immediately.

If the database is in `ARCHIVELOG` mode, then you can make inconsistent backups that are recoverable using archived redo log files. Open database backups are inconsistent because the online redo logs contain changes not yet applied to the datafiles. The redo logs must be archived and then backed up with the datafiles to ensure recoverability.

Despite the name, an inconsistent backup is as robust a form of backup as a consistent backup. The advantage of making inconsistent backups is that you can back up your database while the database is open for updates.

See Also:

- "About Archived Redo Log Files" on page 6-4
- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `BACKUP` command in the RMAN client

Media Recovery

If you restore the archived redo logs and datafiles, then you must perform media recovery before you can open the database. Any database transactions in the archived redo logs not already reflected in the datafiles are applied to the datafiles, bringing them to a transaction-consistent state before the database is opened.

Media recovery requires a control file, datafiles (typically restored from backup), and online and archived redo logs containing changes since the time the datafiles were backed up. Media recovery is most often used to recover from media failure, such as the loss of a file or disk, or a user error, such as the deletion of the contents of a table.

Media recovery can be a complete recovery or a point-in-time recovery. In a complete recovery, you restore backup datafiles and apply all changes from the archived and online redo logs to the datafiles. The database is returned to its state at the time of failure and can be opened with no loss of data.

In a point-in-time recovery, you return a database to its contents at a user-selected time in the past. You restore a backup of datafiles created before the target time and a complete set of archived redo log files from backup creation through the target time. Recovery applies changes between the backup time and the target time to the datafiles. All changes after the target time are discarded.

Oracle Enterprise Manager Database Control (Database Control) provides an interface to both a complete and a point-in-time recovery in the form of a Recovery wizard. However, this guide focuses on complete recovery.

See Also:

- "Performing User-Directed Recovery" on page 9-25
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about complete and point-in-time recovery

Flash Recovery Area

To simplify the management of backup and recovery files, you can create a flash recovery area for your database. The flash recovery area is an Oracle-managed directory, file system, or Automatic Storage Management disk group that provides a centralized disk location for backup and recovery files. Oracle creates archived logs and flashback logs in the flash recovery area. RMAN can store its backup sets and image copies in the flash recovery area, and it uses it when restoring files during media recovery. The flash recovery area also acts as a disk cache for tape.

Oracle Database automatically manages this storage, deleting files that are no longer needed. Periodically copying backups to tape frees space in the flash recovery area for other files. Oracle recommends that you enable a recovery area to simplify backup management. Except as noted, this documentation assumes the use of a flash recovery area.

See Also:

- "Planning Space Usage and Location for the Flash Recovery Area" on page 9-5
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about configuring a flash recovery area

RMAN Repository

RMAN maintains a record of database files and backups for each database on which it performs operations. This metadata is called the **RMAN repository**.

If you back up a file without using RMAN, such as by copying the file at the host operating system level, then you can add metadata about the copy to the RMAN repository. If you later use a command such as `RESTORE DATABASE`, then RMAN uses the records in the repository to select backups needed for the recovery.

The primary location for the RMAN repository for a database is its control file. The importance of this metadata for RMAN is one more reason why protecting your control file is a vital part of your backup strategy. In some installations, a second copy of the RMAN repository is stored in a schema called the **recovery catalog**. The recovery catalog is located in a separate database and can store metadata for multiple databases. Use of a recovery catalog is optional and is beyond the scope of this guide.

See Also:

- "Displaying Backup Reports" on page 9-17
- "Managing Backups" on page 9-18
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about RMAN reporting capabilities and querying the recovery catalog

Configuring Your Database for Basic Backup and Recovery

This section explains how to set up your database to take advantage of Oracle suggested backup strategies. If you already configured the database for automated backups with the Oracle Database Configuration Assistant (DBCA), then skip this section.

To take maximum advantage of Oracle Database features that automatically manage backup and recovery files and operations, configure your database as follows:

- Use a flash recovery area, which automates storage management for most backup-related files, and specify it as an archived redo log destination.
- Run the database in `ARCHIVELOG` mode so you can perform online backups and have data recovery options such as complete and point-in-time media recovery.

You must also set a number of policies governing which files are backed up, what format is used to store backups on disk, and when files become eligible for deletion. See "Using DBCA to Create and Configure a Database" on page 2-8 to learn how to create a database preconfigured for automated daily backups.

Specifying Credentials for Backup and Recovery Using Database Control

You must have the proper credentials to perform some of the configuration tasks for backup and recovery, and to schedule backup jobs and perform recovery. The following credentials may be required:

- The Oracle Database user you specify when you log in to Oracle Enterprise Manager Database Control (Database Control)
- The host operating system user whose credentials you provide when performing backup and recovery tasks

To enter credentials for backup and recovery tasks:

1. Log in to Database Control as a database user with `SYSDBA` privileges, or provide host operating system credentials for a user in the `dba` group on UNIX or Linux, or the `ora_dba` group on Microsoft Windows.

   The host operating system user must also have execute permission for the RMAN command-line client.

   For tasks requiring host operating system credentials, a Host Credentials form appears at the bottom of the page used to perform the task. Enterprise Manager uses the credentials when it starts RMAN jobs that you requested or scheduled.

2. Optionally, in the Host Credentials form, select **Save as Preferred Credential**.

   If you select this option before performing your action, then the provided credentials are stored persistently for the currently logged-in Oracle Database user. The preferred credentials are reused by default whenever you log in as that user and perform operations requiring host credentials.

Note: In situations in which the database is shut down, you may still be prompted for host credentials even if you have saved the preferred credentials.

Planning Space Usage and Location for the Flash Recovery Area

You should place the flash recovery area on a separate disk from the working set of database files. Otherwise, the disk becomes a single point of failure for your database.

The amount of disk space to allocate for the flash recovery area depends on the size and activity levels of your database, which determine the size of your datafiles and redo logs files in addition to your recovery objectives. Your objectives dictate what kinds of backups you use, when you make them, and how long to keep them.

About the Backup Retention Policy and the Flash Recovery Area

Space management in the flash recovery area is governed by a backup retention policy. A retention policy determines when files are obsolete, meaning that they are no longer needed to meet your data recovery objectives.

Retention policies can be based on redundancy of backups or on a recovery window (period of time). When using a policy based on redundancy, the flash recovery area considers a backup of a file obsolete only when the RMAN repository has records of a specified number of more recent backups of that file. When using a recovery policy based on a period of time (or window), you specify a time interval in days. Files are obsolete only when they are no longer needed for complete recovery or point-in-time recovery to a system change number (SCN) within the window. Therefore, a recovery retention policy based on a window is recommended.

Even after files in the flash recovery area are obsolete, they are typically not deleted until space is needed for new files. As long as space permits, files recently moved to tape remain on disk to avoid restoring them from tape for a recovery. The automatic deletion of obsolete files and files moved to tape from the flash recovery area makes it a convenient archiving destination. Other destinations require manual deletion of logs.

About the Flash Recovery Area Size

Oracle Database Backup and Recovery User's Guide explains how to size the flash recovery area. As a general rule, the larger the flash recovery area, the more useful it is. Ideally, the flash recovery area should be large enough for copies of the datafiles, control files, online redo logs, and archived logs needed to recover the database, and also the copies of these backup files that are kept based on the retention policy.

If your backup strategy includes incremental backups, which are described in "Incremental Backups of Datafiles" on page 9-10, then add enough space to the flash recovery area for these files. If you can move some backups to tape, then you can reduce the size of the flash recovery area. Note that retrieving files from tape causes longer database restore operations and recovery times.

Configuring Recovery Settings

In the Recovery Settings page, you can configure settings for instance recovery, media recovery, and flash recovery. In this section, you configure the flash recovery area and enable archiving for the database.

You can configure a flash recovery area when first creating the database. If you did not perform this task at database creation time, however, then you can create a flash recovery area for your database now.

To configure a flash recovery area and put the database in `ARCHIVELOG` mode:

1. On the host operating system, create a directory to hold the flash recovery area.

   Make sure that the operating system permissions for this directory allow the database to create files.

2. On the Database Home page, click **Availability** to display the Availability subpage.

3. In the Backup/Recovery section, click **Recovery Settings**.

   The Recovery Settings page appears.

4. Complete the following steps:

   a. In the Media Recovery section, select **ARCHIVELOG Mode**.

   b. If `USE_DB_RECOVERY_FILE_DEST` is not already set as an archiving destination, then set it.

      This initialization parameter indicates that the flash recovery area should be an archiving destination.

      For ease of database management, the best practice is to use the flash recovery area as your *only* archiving destination.

   c. In the Flash Recovery section, enter the path to the flash recovery area created in Step 1 in **Flash Recovery Area Location**, and select a value for **Flash Recovery Area Size**.

   d. Select **Enable Flashback Database**.

      This option specifies that the database should generate flashback logs in the flash recovery area, enabling you to use Flashback Database. During usual operation, the database occasionally logs images of data blocks to the flashback logs. The database automatically creates, deletes, and resizes flashback logs.

   e. Ensure that the **Apply changes to SPFile only** box is not selected.

f. Click **Apply** to save your changes.

A message prompts you to restart the database.

5. Click **Yes**.

The Restart Database: Specify Host and Target Database Credentials page appears.

6. Enter your host and database credentials, and then click **OK**.

The Restart Database: Confirmation page appears.

7. Click **Yes** to begin the restart operations.

You can click **Refresh** periodically to monitor the progress of the operation.

8. Make a consistent (that is, offline) backup of your entire database *immediately* after switching your database to `ARCHIVELOG` mode.

Caution: You cannot use backups from *before* the switch to `ARCHIVELOG` mode to restore and recover the database to a point in time *after* the switch. Thus, if you do not *immediately* make a backup after switching, you are running your database without a backup. See "Performing and Scheduling Backups Using Database Control" on page 9-11 to learn how to make database backups.

See Also:

- "Rewinding a Database Using Oracle Flashback Database" on page 9-30
- *Oracle Database Backup and Recovery User's Guide* for more details about configuring a flash recovery area

Monitoring Flash Recovery Area Usage

It is important to monitor space usage in the flash recovery area to ensure that it is large enough to contain backups and other recovery-related files.

To monitor available space in the flash recovery area:

- In the High Availability section of the Database Home page, click the link next to **Usable Flash Recovery Area (%)**.

  The Recovery Settings page appears.

  The Reclaimable Flash Recovery Area (GB) and Free Flash Recovery Area (GB) settings indicate how much space is available.

Configuring Backup Settings

You can configure a number of backup-related settings and policies. For example, you can determine how backups are stored, which data is backed up, and how long backups are retained. You can also configure settings to optimize backup performance.

This section explains the concepts behind the available settings and also explains how to change them using the Backup Settings page in Database Control. The settings on the Device subpage affect how RMAN writes backups to disk and to tape.

About RMAN Backups

Database backups created by RMAN are stored as image copies or backup sets.

Image copies are exact byte-for-byte copies of files. You can create an image copy by copying a file at the operating system level. Unlike copying files at the operating system level, however, image copies created through RMAN or Database Control are recorded in the RMAN repository so that RMAN can use these copies during database restore operations and recovery. RMAN can restore files only if they are recorded in the RMAN repository. RMAN can create image copies only on disk.

Backup sets are logical entities produced by the RMAN `BACKUP` command. This command can produce one or more backup sets on disk or media management devices. RMAN can write backup sets only to a media manager.

Each backup set contains several physical files called **backup pieces**. A backup piece stores the backup of one or more database files in a compact RMAN-specific format. One advantage of backup sets is that RMAN uses unused block compression to save space in backing up datafiles. Only those blocks in the datafiles that have been used to store data are included in the backup set.

RMAN depends on server sessions, processes that run on the database server, to create backups and restore them. Each server session, in turn, corresponds to an RMAN channel, representing one stream of data to or from a backup device. Channels are either of type disk or type SBT (tape).

RMAN supports parallelism, which is the use of multiple channels and server sessions to carry out the work of one backup or recovery task. Proper use of parallelism can greatly increase performance on backup and recovery tasks.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for an overview of RMAN architecture
- *Oracle Database Backup and Recovery User's Guide* for more conceptual details about RMAN backups

Configuring Backup Device Settings

For disk-based backups, you can configure the default format for backups, the location on disk where backups are stored, and whether or not backup tasks run in parallel.

For tape backups, you can configure settings such as the number of tape drives and whether or not backups are compressed. On most platforms, you must integrate a media manager with the Oracle database to use sequential media for storage.

You can use Oracle Secure Backup, which supports both database and file system backups to tape, as your media manager. Oracle Secure Backup provides the same services for RMAN as other third-party SBT interfaces, but is better integrated with Database Control. This section assumes that you will make only disk backups.

To configure backup settings for disk:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Backup Settings**.

   The Backup Settings page appears.
3. Click **Device**.

   The Device subpage of Backup Settings appears.
4. Complete the following steps:

   a. In **Parallelism**, enter **1**.

Later, when you have had time to review the information in *Oracle Database Backup and Recovery User's Guide* about parallelism and performance in RMAN, you may want to change this value.

b. Leave the Disk Backup Location field blank so that RMAN directs backups to the flash recovery area.

c. In the **Disk Backup Type** section, select **Backup Set**.

One advantage of backup sets is that RMAN uses unused block compression to save space when backing up datafiles. Only those blocks in the datafiles that have been used to store data are included in the backup set.

5. In Host Credentials, enter values in the **Username** and **Password** fields.

Host credentials are described in "Specifying Credentials for Backup and Recovery Using Database Control" on page 9-4.

6. Click **Test Disk Backup** to ensure the credentials and backup location are correct.

A message appears that states whether or not the test was successful.

In this example you do not change the settings on the Backup Set subpages.

See Also:

- *Oracle Secure Backup Administrator's Guide* to learn how to set up and use Oracle Secure Backup with Database Control
- *Oracle Database Backup and Recovery User's Guide* to learn how to use the RMAN client to configure the default device for backups

Configuring Backup Policy Settings

You can set the backup policies that govern control file and server parameter file backups, tablespaces to exclude from an entire database backup, and the backup retention policy.

To configure the backup policy settings:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Backup Settings**.

The Backup Settings page appears.

3. Click **Policy**.

The Backup Policy subpage appears.

4. Perform the following actions:

- Select the **Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change** option. Leave the Autobackup Disk Location field blank so that RMAN stores the automatic backups in the flash recovery area.

  The server parameter file and control file are critical to the database and RMAN, and are also relatively small compared to typical datafiles. Backing them up frequently results in relatively little storage overhead.

- Select the **Optimize the whole database backup by skipping unchanged files such as read-only and offline datafiles that have been backed up** option.

  This option saves space in the flash recovery area.

- Select the **Enable block change tracking for faster incremental backups** option. Either leave the Block Change Tracking File field blank (if you configured a database area in "Step 7 - Database File Locations" on page 2-11) or enter a file name.

  This option takes advantage of the block change tracking feature, which greatly improves incremental backup performance at a small cost of overhead.

5. In the Tablespaces Excluded From Whole Database Backup section, leave the settings as they are.

   This feature enables you to specify a list of tablespaces to exclude from a backup. For example, you do not need to include read-only tablespaces in every backup.

6. In the Retention Policy section, select **Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery)**. In **Days**, enter **`31`**.

   This setting enables a recovery policy with retention based on a period of time.

7. In the Archivelog Deletion Policy section, select **None**.

   This option specifies that logs are eligible for automatic deletion only when they have been backed up to tape or are obsolete based on the retention policy.

8. Click **OK** to save your changes.

See Also:

- "Managing Backups" on page 9-18
- *Oracle Database Backup and Recovery User's Guide* to learn how to configure a backup retention policy

Backing Up Your Database

This section describes how to back up your database with Oracle Enterprise Manager Database Control (Database Control). The Oracle suggested strategy for disk-only backups provides efficient daily backup of the database. This strategy enables you to quickly return your database to its state at any point during the previous 24 hours. For more flexible backup options, see *Oracle Database Backup and Recovery User's Guide*.

RMAN Backup Concepts

This section provides the concepts you need to understand the Oracle suggested backup strategy and other backup types provided through Database Control.

Full Backups of Datafiles

A full backup of a datafile includes all used blocks of the datafile. As explained in "About RMAN Backups" on page 9-7, an RMAN backup can be either an image copy or backup set. An image copy is a bit-for-bit copy of the datafile, and thus includes unused blocks.

Incremental Backups of Datafiles

RMAN incremental backups copy only those blocks in a datafile that change between backups. A level 0 incremental backup, which copies all blocks in the datafile, is used as a starting point for an incremental backup strategy.

Incremental backups at level 1 copy only images of blocks that have changed since a previous incremental backup. Level 1 backups can be *cumulative*, in which case all

blocks changed since the most recent level 0 backup are included, or *differential*, in which case only those blocks changed since the most recent level 0 or level 1 incremental backup are included. A typical incremental strategy makes level 1 backups at regular intervals such as once each day.

Recovering changed blocks from incremental backups can speed media recovery. Because an incremental level 1 backup copies the final contents of all datafile blocks changed during the period covered by the incremental backup strategy, the recovery process can omit reapplying individual updates from the redo logs of that period, and update each block with its final contents. The redo logs are used only for changes from the period not covered by level 1 backups.

Incrementally Updated Backups: Rolling Forward Image Copies of Datafiles

RMAN enables you to apply level 1 incremental backups to an older image copy of your datafiles. You can roll forward the old copy to the point in time of the most recent level 1 incremental backup. All blocks changed since the image copy was created are overwritten with their new contents as of the time of the last level 1 incremental backup. The effect is to roll the file forward in time, so that its contents are equivalent, for the purposes of database recovery, to an image copy of the datafile made at the time of the last incremental level 1 backup.

Incorporating incrementally updated backups into your backup strategy shortens expected recovery times. The reason is that media recovery to the present time or to a point in time in the recent past can begin at the time of the last level 1 backup applied, rather than at the time of the last full database backup.

Backup Tags

All RMAN backups, including incremental backups, are labeled with a tag. A **tag** is a text string that identifies that backup, either uniquely or as part of a group of backups. For instance, if you performed a weekly, full database backup on Saturday nights, you could use the tag `FULL_SAT` to identify all such backups.

You can use tags to refer to specific backups in RMAN commands. For example, you could issue a command to move the latest `FULL_SAT` backup to tape. If you do not specify a tag, then RMAN creates a unique tag automatically.

Because you can use tags to refer to different groups of backups, you can create different routines in your backup strategy that do not interfere with each other. When you schedule a backup job and give the job a name, the job name is used as a tag.

Performing and Scheduling Backups Using Database Control

Database Control enables you to perform RMAN backups and schedule backup jobs required by your backup strategy.

Performing a Whole Database Backup

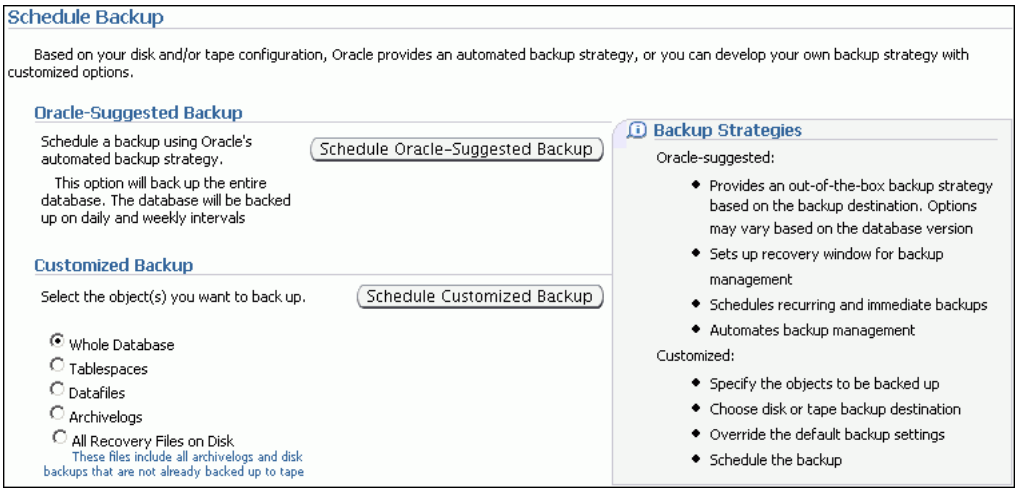
Whole backups of a database include the complete contents of all datafiles of the database, as well as the control file, archived redo logs, and server parameter file. With these files, you can perform a complete recovery.

While whole database backups can be an important element in your overall backup strategy, they are also a required step in some situations, such as when you enable or disable `ARCHIVELOG` mode (see "Configuring Recovery Settings" on page 9-6). This section explains how to make whole database backups to disk.

To perform a whole database backup when the database is open:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section of the Availability page, click **Schedule Backup**.

   The Schedule Backup page appears.



3. In the Customized Backup section, select **Whole Database**, and then click **Schedule Customized Backup**.

   The Schedule Customized Backup: Options page appears.
4. In the Backup Type section, select **Full Backup**.
5. In the Backup Mode section, select either **Online Backup** or **Offline Backup**.

   Typically, you will want to perform online backups to maximize database availability.

   Note: As explained in "Configuring Your Database for Basic Backup and Recovery" on page 9-4, you can make online backups only if your database is running in `ARCHIVELOG` mode.

6. In the Advanced section, make the following selections, and then click **Next**:
   - If you are performing an online backup, select **Also back up all archived logs on disk**.

     You do not need to back up archived logs when performing an offline backup because the database is in a consistent state at the time of backup and does not require media recovery if you restore the database from this backup. However, you can choose to include archived logs in the backup.
   - If a flash recovery area is your only archiving destination, do not select "Delete all archived logs from disk after they are successfully backed up".

     In this case, redo logs that have been backed up are deleted automatically as space is needed. If you are using some other archiving destination, it may be useful to select this option as part of managing your backup storage.
   - Do not select "Use proxy copy supported by media management software to perform a backup" for now.

- If the flash recovery area stores backups, do not select the Delete obsolete backups option.

  In this case, obsolete backups are deleted automatically as space is needed. If you use another destination for your backup files, you can select this option.
- Do not enter a value for Maximum Files per Backup Set.
- Do not select the Encryption option.

  For improved security, you can encrypt RMAN backup sets with a user-selected algorithm. Encrypted backups cannot be read by unauthorized users.

The Schedule Customized Backup: Settings page appears.

7. Select **Disk** and then click **Next**.

   Oracle recommends backing up to disk when possible to minimize recovery time by minimizing restore operations from tape. Later, you can move disk backups to tape.

   The Schedule Customized Backup: Schedule page appears.

8. Complete the following steps:

   a. In the Job section, leave the default values for **Job Name** and **Job Description**.

      In the **Job Name** field, you can enter a user-specified tag for this backup. The job name is used as a prefix for the backup tag for backups created by this job.

      Note: For recurring jobs, it is useful to set a **Job Name** so that the resulting backups are easy to identify as part of an ongoing series. Use a descriptive tag, such as `WEEKLY_FULL_BACKUP`.

      In the **Job Description** field, you can enter any descriptive text that is useful for your own reference.

   b. In the Schedule section, click **One Time (Immediately),** and then click **Next**.

      The Schedule Customized Backup: Review page appears.

9. Click **Submit Job**.

   The Status page appears. This page should contain a message that indicates the job was successfully submitted.

10. Click **View Job** to monitor the progress of the backup.

    The Execution: Database page appears. This page contains a Summary section describing the job. The table at the bottom of the page displays the progress of the various steps of the backup job. You can reload this page in your browser to monitor the ongoing progress of the job.

    In the Name column of the table, you can see the current phase of the RMAN job. Clicking the name of the phase of the backup job displays a page containing the RMAN output for that part of the job. From this page, click the **Back** button in your browser to return to the Execution: Database page.

To perform a whole database backup when the database is closed:

- Follow the same steps as for an open database backup, except select **Offline Backup** in Step 4 and then continue with the steps that follow.

Using the Oracle Suggested Backup Strategy

Database Control makes it easy to set up an Oracle suggested backup strategy for backups to disk that protects your data and provides efficient recoverability to any point in the user-specified recovery window (time period). The Oracle suggested strategy leverages the incremental backup and incrementally-updated backup features to provide faster backups than whole database backups, and faster recoverability than is possible through applying database changes from the archive log to your datafiles.

About the Oracle Suggested Backup Strategy The Oracle suggested strategy is based on creating an image copy of your database. The database rolls forward this copy by using incrementally updated backups. Oracle Enterprise Manager schedules RMAN backup jobs during the overnight hours.

For each datafile, the strategy calls for backups as follows:

- At the beginning of day 1 of the strategy (the time the first scheduled job actually runs), RMAN creates an incremental level 0 datafile copy. It contains the datafile contents at the beginning of day 1.

  If a recovery is required, the redo logs from day 1 can be used to recover to any point during day 1.

- At the beginning of day 2, RMAN creates an incremental level 1 backup that contains the blocks changed during day 1.

  If a recovery is required, RMAN can apply this incremental level 1 to roll forward the level 0 backup to the beginning of day 2. RMAN can use redo logs to recover to any point during day 2.

- At the beginning of each day *n* for day 3 and onward, RMAN applies the level 1 backup from the beginning of day *n*-1 to the level 0 backup. This action brings the datafile copy to its state at the beginning of day *n*-1. Then, RMAN creates a new level 1 backup that contains the blocks changed during day *n*-1.

  If a recovery is required, RMAN can apply this incremental level 1 backup to the datafile rolled forward on day *n*-1 to the beginning of day *n*. RMAN can use redo logs to recover the database to any point during day *n*.

The datafile copies used in the Oracle suggested backup strategy are tagged as `ORA$OEM_LEVEL_0`. The level 1 incremental backups for use in this strategy are created for use with datafile copies with this label. You can safely implement other backup strategies without interfering with the Oracle suggested backup strategy.

Oracle suggested backup strategies also use tape backups in addition to disk backups, but these are beyond the scope of this section.

Backing Up Your Database Using the Oracle Suggested Disk Backup Strategy The following procedure schedules daily database backups. The strategy uses incremental backups and incrementally applied backups to enable quick recovery to any time in the preceding 24 hours.

To backup your database using the Oracle-suggested disk backup strategy:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section of the Availability page, click **Schedule Backup**.

   The Schedule Backup page appears.
3. In the Oracle-Suggested Backup section, click **Schedule Oracle-Suggested Backup**.

The Schedule Oracle-Suggested Backup: Destination page appears. On this page, you select the destination media for the backup, which can be disk, tape, or both.

4. Select **Disk** and then click **Next**.

   The Schedule Oracle-Suggested Backup: Setup page appears.

   This page describes the backups that are performed each day as part of the disk-based strategy.

5. Review the information and then click **Next**.

   The Schedule Oracle-Suggested Backup: Schedule page appears.

6. Choose a **Start Date**, **Time Zone**, and **Daily Backup Time** for the daily backups, and then click **Next**.

   The Schedule Oracle-Suggested Backup: Review page appears.

   Database Control displays the backup script that RMAN will run (although you cannot edit the script directly). The script should look similar to the following:

```
Daily Script:
run {
allocate channel oem_disk_backup device type disk;
recover copy of database with tag 'ORA$OEM_LEVEL_0';
backup incremental level 1 cumulative  copies=1 for recover of copy with tag
'ORA$OEM_LEVEL_0' database;
}
```

7. Click **Submit Job**.

Scheduling Miscellaneous Backup Tasks

After taking some time to understand the full range of available backup options described in *Oracle Database Backup and Recovery Basics*, you may decide to schedule backup tasks beyond those used to implement the Oracle suggested backup strategy.

To schedule miscellaneous backup tasks:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section of the Availability page, click **Schedule Backup**.

   The Schedule Backup page appears.

3. In the Customized Backup section, select the object that you want to back up. For example, select **Tablespaces** or **Archivelogs**.
4. Click **Schedule Customized Backup** to start the Schedule Customized Backup wizard.
5. On the following pages of the wizard, specify details such as objects to back up, and required options and settings. Click **Next** to proceed to the next page.

   The choices presented on these pages are determined by the type of objects being backed up. For more information about using the wizard, click **Help** on any page.

6. Click **Submit Job**.

Validating Backups and Testing Your Backup Strategy

As part of your backup strategy, you should periodically check whether or not your backups are intact and can be used to meet your recoverability objectives. You can validate your backups in the following ways:

- Select specific backup sets or image copies in Database Control and validate them. This technique indicates if a backup exists and can be restored. For this form of validation, use the Manage Current Backups pages as described in "Validating Selected Backups" on page 9-16.
- Specify database files and let RMAN select backups to use when restoring those files, as it would for an actual restore operation. This technique ensures that your available backups are sufficient to restore the database. For this form of validation, use the Perform Recovery pages as described in "Validating Backups for Restore Operations" on page 9-17.

> **Note:** Validating backups stored on tape can be time-consuming because the entire backup is read from tape.

You can set up both forms of validation as scheduled tasks in Database Control. You should incorporate both forms of validation into your backup strategy to ensure that your recoverability goals are met by your available backups.

Validating Selected Backups

Validating specific backups checks whether or not these backups exist and can be restored. It does not test whether or not the set of available backups meet your recoverability goals. For example, image copies of datafiles for several tablespaces from your database may exist, each of which can be validated. If there are some tablespaces for which no valid backups exist, however, then you cannot restore and recover the database.

To validate selected backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, click **Manage Current Backups**.

   The Manage Current Backups page appears.
3. Use the search features on this page to identify the backups whose contents you want to validate.
4. Select the check boxes next to each backup of interest in the list of current backups, and click **Validate**.

   The Validate: Specify Job Parameters page appears.
5. Specify a job name and description, and time settings for starting and repeating the operation, and then click **Submit Job**.

   A message confirms the submission of the job.
6. Click **View Job** to display details about the validation.

See Also:

- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `VALIDATE` command

Validating Backups for Restore Operations

You can test whether or not a sufficient set of backups exists that can be used to restore the specified database files. After you specify which tablespaces to restore and, possibly, a time as of which to restore them, RMAN selects a set of backups that contain the needed data. RMAN reads the selected backups in their entirety to confirm that they are not corrupt, but does not produce output files.

Validating the restoration of files tests whether or not the file can be restored given the available backups, but it does not test whether or not all backups of the specified object are valid.

To verify whether or not specified database files can be restored:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, click **Perform Recovery**.

   The Perform Recovery page appears.
3. In the User-Directed Recovery section, select **Datafiles** or **Tablespaces** from the Recovery Scope list, depending on what you want to validate.
4. For the Operation Type, select **Restore Datafiles** or **Restore Tablespaces** and then click **Recover**.

   The Perform Object Level Recovery page appears.
5. Click **Add** to add tablespaces or datafiles for the validate operation. After making your selections, click **Next**.

   The Perform Object Level Recovery: Restore page appears.
6. In the Backup Selection section, specify which backups should be restored.
7. In the Backup Validation section, select **Validate the specified backup without restoring the datafiles**. Then, click **Next**.

   The Perform Object Level Recovery: Review page appears.
8. Click **Submit Job** to run the validation.

   The Perform Recovery: Result page appears.

See Also:

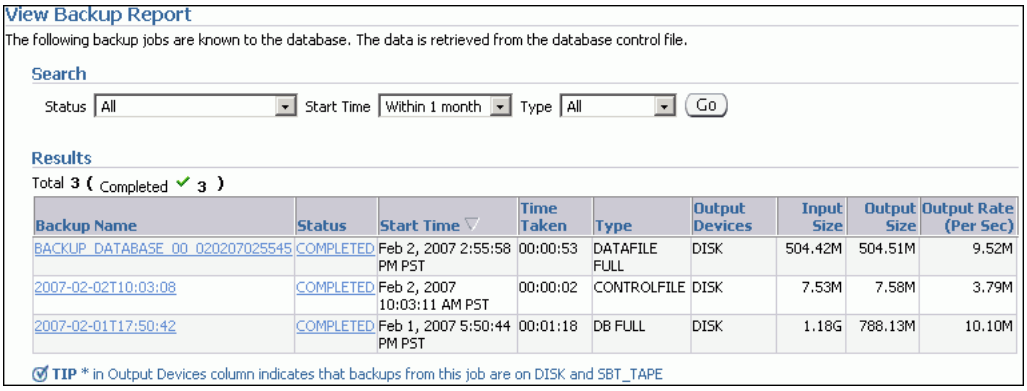
- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `RESTORE ... VALIDATE` command

Displaying Backup Reports

Backup reports contain summary and detailed information about past backup jobs run by RMAN, including both backups run through Oracle Enterprise Manager Database Control (Database Control) and the RMAN command-line client.

To display backup reports:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, in the Backup and Recovery section, click **Backup Reports**.



The Backup Reports page contains a list of recent backup jobs. You can use the Search section of the page to restrict (filter) the backups listed by the status of the job, the start time of the backup, and the type. In the Search sections, specify any filter conditions and click **Go**.

3. Click the link in the **Backup Name** column to display detailed information about any backup.

   The Backup Report page is displayed for the selected backup.

4. Click the link in the **Status** column to view a log of RMAN output for the job.

Note: The control file view `V$RMAN_OUTPUT` contains the output of recent RMAN jobs. The contents of this view are not preserved when the instance is restarted. Therefore, the output from past RMAN jobs may not be available.

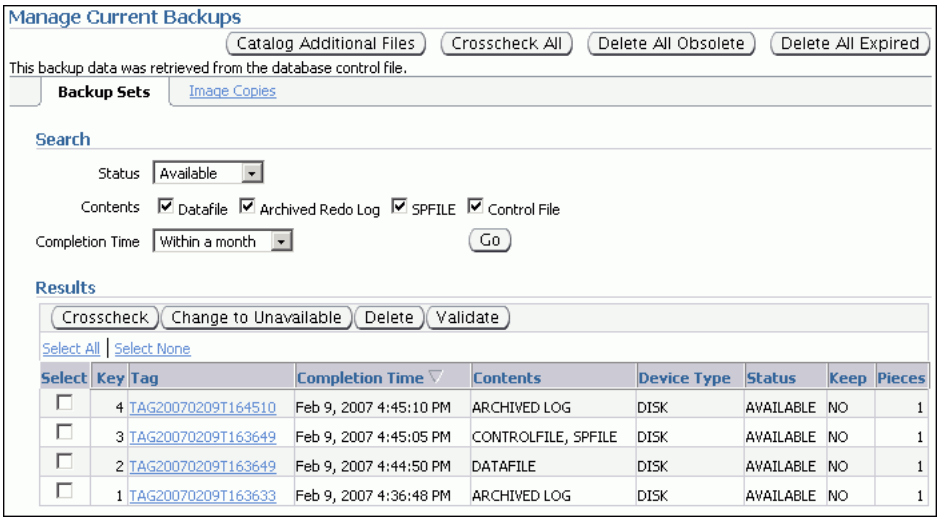
Managing Backups

As part of a backup strategy, you need to manage database backups. A related task is managing the record of those backups in the RMAN repository. Oracle Enterprise Manager Database Control (Database Control) simplifies these tasks.

About Backup Management

An essential part of a backup and recovery strategy is managing backups after you create them. Backup management includes deleting obsolete backups and performing periodic checks to ensure that backups are available and usable.

You can perform backup management tasks from the Manage Current Backups page. This page has two subpages: Backup Sets and Image Copies. Both pages have similar purposes: listing backups based on the record in the RMAN repository and enabling you to manage the backups.



A backup recorded in the RMAN repository has one of the following status values:

- Available, meaning that the backup is still present on disk or tape, as recorded in the repository
- Expired, meaning that the backup no longer exists on disk or tape, but is still listed in the repository
- Unavailable, meaning that the backup is temporarily not available for data recovery operations (because, for example, it is stored on a tape that is stored offsite or on a disk that is currently not mounted)

Backups can also be obsolete. An obsolete backup is, based on the currently configured retention policy, no longer needed to satisfy data recovery goals.

Maintenance tasks that you can perform in Database Control include the following:

- Viewing details about your backups
- Cross-checking your repository, which means checking whether or not backups listed in the repository exist and are accessible, and marking as expired any backups not accessible at the time of the cross-check
- Deleting the record of expired backups from your RMAN repository
- Deleting obsolete backups from the repository and from the backup media
- Validating backups to ensure that a given backup is available and not corrupted

> **Note:** If a backup no longer exists, immediately delete the backup record from the RMAN repository. Without an accurate record of available backups, you may discover that you no longer have complete backups of your database when you need to perform a recovery.

As with the backup and restore commands in Database Control, the commands to cross-check, delete, and change the status of backups are translated to RMAN commands. The commands are submitted as RMAN jobs that can be run immediately or be scheduled. Some tasks, such as periodic cross-checks of your backups, should be among the regularly scheduled components of your backup strategy.

If you use a flash recovery area for backup storage, then many maintenance activities are reduced or eliminated. The automatic disk space management mechanisms delete backups and other files as needed, thereby satisfying disk space demands for ongoing database operations without compromising the retention policy.

Cross-Checking Backups

Cross-checking a backup synchronizes the physical reality of backups with their logical records in the RMAN repository. For example, if a backup on disk was deleted with an operating system command, then a cross-check detects this condition. After the cross-check, the RMAN repository correctly reflects the state of the backups.

Backups to disk are listed as available if they are still on disk in the location listed in the RMAN repository, and if they have no corruption in the file header. Backups on tape are listed as available if they are still on tape. The file headers are not checked for corruption. Backups that are missing or corrupt are listed as expired.

To cross-check individual files:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Manage Current Backups**.

   The Manage Current Backups page appears.
3. Search for the backup sets or image copies whose contents you want to cross-check.
4. In the Results section, select each backup to be included in the cross-check operation.

   Database Control does not support selecting both image copies and backup sets for a cross-check within a single operation.
5. Click **Crosscheck** at the top of the Results list.

   After a confirmation page is displayed, Database Control performs the cross-check.

To cross-check all files:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Manage Current Backups**.

   The Manage Current Backups page appears.
3. In the Manage Current Backups page, click **Crosscheck All**.

   The Crosscheck All: Specify Job Parameters page appears. You can schedule the cross-check to run now or later, or specify regularly scheduled cross-checks.
4. Click **Submit Job**.

Note: Cross-checking all backups in the RMAN repository may take a long time, especially for tape backups. A cross-check of all files, unlike cross-checking individual files, is handled as a scheduled job.

See Also:

- "Managing Backups"
- *Oracle Database Backup and Recovery User's Guide* for details on how to perform this procedure using the RMAN `CROSSCHECK` command

Deleting Expired Backups

Deleting expired backups removes from the RMAN repository those backups that are listed as `EXPIRED`. Expired backups are those found to be inaccessible during a cross-check. No attempt is made to delete the files containing the backup from disk or tape; this action updates only the RMAN repository.

To delete expired backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Manage Current Backups**.

   The Manage Current Backups page appears.
3. In the Manage Current Backups page, click **Delete All Expired**.

   This action deletes expired backup sets and expired image copies from the RMAN repository, regardless of which subpage is currently active.

   The Delete All Expired: Specify Job Parameters page appears.
4. Optionally, select **Perform the operation 'Crosscheck All' before 'Delete All Expired'**.

   By performing the cross-check immediately before deleting expired backups, RMAN will have up-to-date information about which backups are expired.
5. Click **Submit Job**.

   A message should appear indicating that your job was successfully submitted.

Marking Backups as Available or Unavailable

If one or more specific backups are unavailable because of a temporary condition, such as a disk drive that is temporarily offline or a tape stored offsite, then you can mark those backups as unavailable. RMAN does not use unavailable backups to restore or recover data.

> **Note:** Backups stored in the flash recovery area cannot be marked as unavailable.

RMAN keeps the record of unavailable backups in the RMAN repository and does not delete backups listed as unavailable when you delete expired backups. If the unavailable backups become accessible again, then you can mark them as available.

To mark backups as available or unavailable:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Manage Current Backups**.

The Manage Current Backups page appears.

3. In the Search section, find the backups whose status you want to change.
4. Click the **Select** check box next to each backup in the Results list of backups.
5. Do one of the following:
   - Select **Change to Available**.
   - Select **Change to Unavailable**.

   Note: If you restricted the backups listed by searching only for available backups, only the **Change to Unavailable** button appears. If you restricted the backups listed by searching only for unavailable backups, only the **Change to Available** button appears.

   A confirmation message appears.
6. Click **Yes** to perform the change operation.

Deleting Obsolete Backups

This section explains how to delete obsolete backups, which are those no longer needed by the configured retention policy. If you use a flash recovery area as your only disk-based backup destination, then you never need to delete obsolete backups from disk. The flash recovery area will keep files as specified by the retention policy, and delete them only when space is needed.

To delete obsolete backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Manage Current Backups**.

   The Manage Current Backups page appears.
3. Click **Delete All Obsolete**.

   All obsolete backups (both backup sets and image copies) will be deleted, regardless of whether you clicked **Delete All Obsolete** while viewing the Backup Sets or Image Copies subpage.

   The Delete All Obsolete: Specify Job Parameters page appears.
4. In the Schedule section, do one of the following:
   - Select **One Time (Immediately)** to run the deletion job immediately.
   - Schedule the deletion as you would a backup job.
5. Click **Submit Job**.

Performing Oracle Advised Recovery

The Oracle advised recovery feature uses Data Recovery Advisor, which is an Oracle Database feature that automatically diagnoses data failures, determines and presents appropriate repair options, and performs repairs if requested by the user. By providing a centralized tool for automated data repair, Data Recovery Advisor improves the manageability and reliability of an Oracle database.

Note: To recover an Oracle RAC database using Data Recovery Advisor you must use the RMAN interface. You cannot use Enterprise Manager.

About Data Recovery Advisor

In the context of Data Recovery Advisor, a health check is a diagnostic procedure run by the Health Monitor to assess the state of the database or its components. Health checks are invoked reactively when an error occurs. You can also invoke checks manually.

A **failure** is a persistent data corruption detected by a health check. Failures are usually detected reactively. A database operation involving corrupted data results in an error, which automatically invokes a health check in the database. The check searches the database for failures related to the error. If failures are diagnosed, then they are recorded in the Automatic Diagnostic Repository (ADR).

You can use Data Recovery Advisor to generate repair advice and repair failures only after failures have been detected by the database and stored in ADR. Data Recovery Advisor can report on and repair failures such as inaccessible files, physical and logical block corruptions, and I/O failures. Every failure has a failure priority: CRITICAL, HIGH, or LOW. Every failure also has a failure status of OPEN or CLOSED.

You can also use Data Recovery Advisor to view repair options. A **repair** is an action that fixes one or more failures. Examples of repairs include block media recovery, datafile media recovery, and Oracle Flashback Database. Typically, Data Recovery Advisor presents both automated and manual repair options. If appropriate, you can choose an automated repair option in order to perform a repair. In this case, Data Recovery Advisor verifies the repair success, and closes the relevant repaired failures.

Using Data Recovery Advisor

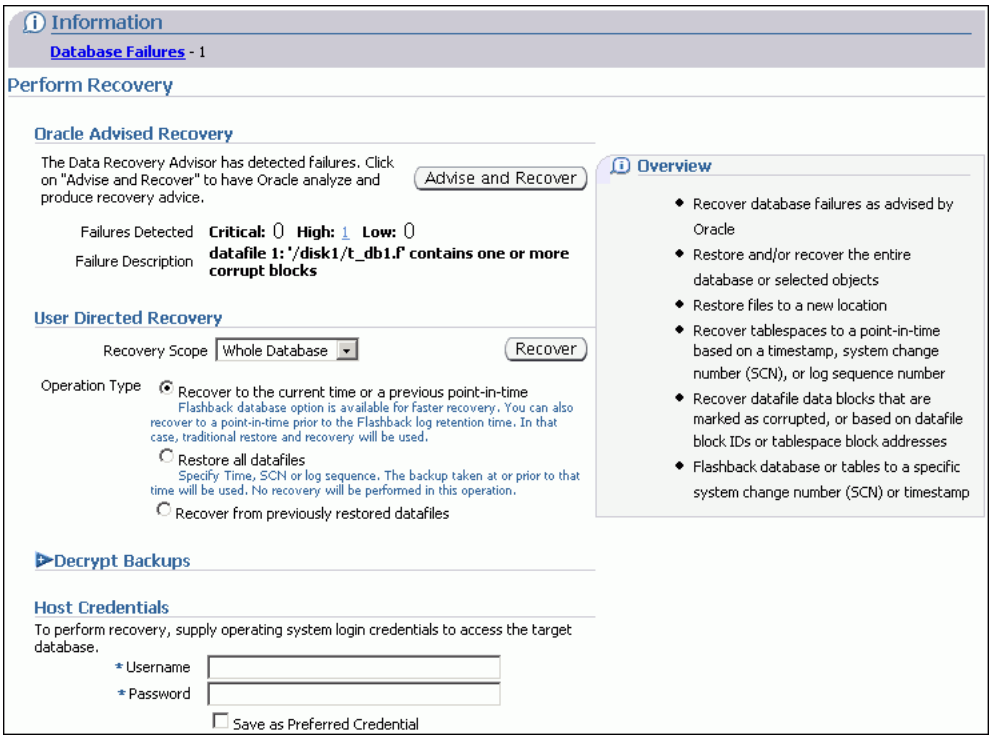
The recovery process begins when you either suspect or discover a failure. You can discover failures in many ways, including error messages, alerts, trace files, and health checks. You can then use Data Recovery Advisor to gain information and advice about failures and repair them automatically.

This section describes a scenario in which you use Data Recovery Advisor to repair a corrupted block. Assume that the Health Meter on the Database Home page indicates that an incident has occurred. The Alerts section of the page indicates that a block corruption has occurred.

To use the Oracle advised recovery strategy:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Click **Perform Recovery**.

   The Perform Recovery page appears. The Perform Recovery page is divided into two sections: Oracle Advised Recovery and User-Directed Recovery. The Oracle Advised Recovery section uses Data Recovery Advisor to automate recovery.



3. Look for failures in the Oracle Advised Recovery section.

   In the screenshot shown in Step 2, the section indicates that one failure with high priority exists. The failure indicates that datafile 1 contains corrupt blocks.

4. Do one of the following:

   - Click **Advise and Recover**.
   - Click the number next to the failure status.

   The View and Manage Failures page appears.

5. In the Priority list, select **All** and then click **Go**.

   This action enables you to view all failures known to Data Recovery Advisor. The data failures are represented as a navigation tree at the bottom of the page.

6. Perform the following actions:

   a. Select **Data Failures** from the navigation tree to expand it (if it is not already expanded).

   b. Select the failure to expand it.

   c. Click **Advise**.

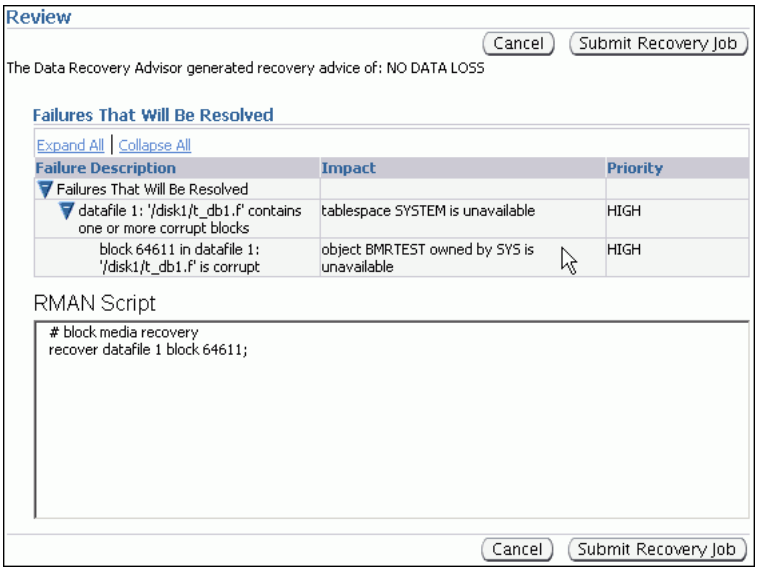
   The Recovery Advice page appears.

   This page shows the RMAN script that will be used to repair the failure. For example, in the case of a corrupted block, the script might look similar to the following:

   ```
   # block media recovery
   recover datafile 1 block 63467;
   ```

7. Click **Continue**.

   The Review page appears. This page summarizes the recovery job.



8. Click **Submit Recovery Job**.

   The Job Activity page appears.

   A table describes details of the repair such as the job name, the job status, the time that the repair is scheduled to be performed, the owner of the job, and so on.

9. Click the name of the repair job and click **View Results**.

   The Job Run page appears.

   You can click the job step in the navigation tree to view the results of the job.

Performing User-Directed Recovery

Oracle Enterprise Manager Database Control (Database Control) User-Directed Recovery provides a Recovery wizard that enables you to use flashback features and perform restore operations and recovery procedures. For example, you can do the following:

- Repair unwanted changes to database objects with the logical flashback features
- Rewind the entire database with Oracle Flashback Database
- Completely restore and recover the database
- Perform point-in-time recovery of the database or selected tablespaces
- Perform block media recovery of datafiles that have corrupted blocks

Database Control can determine which parts of the database must be restored and recovered, including detecting situations such as corrupted database files before they effect database operations. Database Control guides you through the recovery, prompting you for any required information and performing the specified recovery actions.

This section contains a few typical recovery examples so that you can become familiar with the Perform Recovery page. You can use the Perform Recovery page to access other whole database or object-level recovery features of Database Control.

Rewinding a Table Using Oracle Flashback Table

Oracle Flashback Table enables you to rewind one or more tables back to their contents at a previous time without affecting other database objects. Thus, you can recover from logical data corruptions such as table rows added or deleted accidentally. Unlike point-in-time recovery, the database remains available during the flashback operation.

For this example, you will use Flashback Table on the `employees` table in the `hr` schema. Assume that an erroneous update shortly after October 23, 2005 at 15:30:00 has changed the `lastname` column for all employees to an empty string, and you need to return the original `lastname` values to the table.

Enabling Row Movement on a Table

Before you can use Flashback Table, you must ensure that row movement is enabled on the table to be *flashed back*, or returned to a previous state. Row movement indicates that rowids will change after the flashback occurs. This restriction exists because if rowids before the flashback were stored by an application, then there is no guarantee that the rowids will correspond to the same rows after the flashback.

To enable row movement on a table:

1. On the Database Home page, click **Schema** to display the Schema subpage.
2. Click **Tables** in the Database Objects section.

   The Tables page appears.
3. To find the target table for Flashback Table, do the following:
   a. Enter the schema name in the **Schema** field and, optionally, the table name in the **Object Name** field.
   b. Click **Go** to search for the table.

      When you search for tables, for example, in the `hr` schema, you may need to page through the search results to find your table.
4. Select the table from the list of tables and click **Edit**.

   In this scenario, select `employees`.

   The Edit Table: *table_name* page appears.
5. Click **Options** to go to the Options subpage.
6. Complete the following steps:
   a. Set **Enable Row Movement** to **Yes**.
   b. Click **Apply** to update the options for the table.

   An update message appears.
7. Complete the following steps:
   a. Click **Tables** at the top of the page to return to the search results.
   b. Enable row movement on more tables by repeating Step 1 through Step 6 for each table.

   For this example, you should also enable row movement on the tables `hr.jobs` and `hr.departments`.

Performing a Flashback Table Operation

In this example, you rewind the `hr.employees` table and its dependent tables to a previous point in time.

To perform the Flashback Table operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Select **Perform Recovery** from the Backup/Recovery section.

   The Perform Recovery page appears.
3. In the User-Directed Recovery section, select **Tables** from the Recovery Scope list.

   The page reloads with options appropriate for object-level recovery of tables.
4. For **Operation Type**, choose **Flashback Existing Tables**, and click **Recover**.

   The Perform Object Level Recovery: Point-in-time page appears.
5. Choose the target time for the Flashback Table operation, and click **Next**.

   Note: If you do not know the time at which the unwanted changes occurred, then you can investigate the history of transactions affecting this table by selecting **Evaluate row changes and transactions to decide upon a point in time**. Oracle Flashback Versions Query enables you to review all recent changes to the target table. Use of this feature is beyond the scope of this guide.

   For this example, assume that rows were accidentally inserted 5 minutes ago. Select **Flashback to a timestamp** and enter a time 5 minutes before the present time.

   The Perform Object Level Recovery: Flashback Tables page appears.
6. Enter table names in the **Tables to Flashback** text box and then click **Next**.

   You can enter multiple table names to flash back several tables to the same time. You can also click **Add Tables** and search for more tables. For this example, enter the text **`hr.employees`** in the **Tables to Flashback** text box.

   If your table has other dependent tables, then the Dependency Options page appears. This page asks how dependencies should be handled.
7. Do one of the following, and then click **Next**:
   - Select **Cascade** to flash back any dependent tables.
   - Select **Restrict** to flash back only the target table.
   - Select **Customize** to choose which dependent tables to flash back and which to leave as they are.

   You can click **Show Dependencies** to see which tables will be affected.

   Note: Row movement must be enabled on all affected tables, not just the initial target tables.

   In this example, the `hr.employees` table has dependent tables `hr.jobs` and `hr.departments`, so select **Cascade** and then click **Next**.

The Perform Object Level Recovery: Review page appears.

8. Click **Submit**.

   When the operation is completed, a Confirmation page displays the results. Click **OK** to return to the Database Home page.

Recovering a Dropped Table Using Oracle Flashback Drop

Oracle Flashback Drop enables you to reverse the effects of dropping (deleting) a table, returning the dropped table to the database along with dependent objects such as indexes and triggers. This feature stores dropped objects in a recycle bin, from which they can be retrieved until the recycle bin is purged, either explicitly or because space is needed.

As with Flashback Table, you can use Flashback Drop while the database is open. Also, you can perform the flashback without undoing changes in objects not affected by the Flashback Drop operation. Flashback Table is more convenient than forms of media recovery that require taking the database offline and restoring files from backup.

Note: For a table to be recoverable using Flashback Drop, it must reside in a locally managed tablespace. Also, you cannot recover tables in the `SYSTEM` tablespaces with Flashback Drop, regardless of the tablespace type.

Dropping a Table

For the purpose of learning about Flashback Drop, you will create a new table named `reg_hist` and then drop it. The database will place the table in the recycle bin so that it can be retrieved with the Flashback Drop feature.

To create and then drop a table:

1. On the Database Home page, click **Schema** to display the Schema subpage.
2. Click **Tables**.

   The Tables page appears.
3. Enter **`hr`** in the **Schema** field and **`regions`** in the **Object Name** field. Click **Go**.

   The schema and table are listed in the table at the bottom of the page.
4. In the Actions list, select **Create Like** and click **Go**.

   The General subpage of the Create Table page appears.
5. Complete the following steps:
   a. In the **Name** field, enter **`reg_hist`**.
   b. Deselect **Not Null** for the **`REGION_ID`** column.
   c. Click **Constraints** to open the Constraints subpage.

      The Constraints subpage appears.
   d. Select each constraint and click **Delete**.
   e. Click **OK** to create the table.

      A confirmation message appears.

6. In the **Object Name** field, enter **`reg_hist`** and click **Go**.

   The Tables page displays information about this table.

7. Click **Delete with Options** to delete the table.

   The Delete With Options page appears.

8. Select **Delete the table definition, all its data, and dependent objects (DROP)**, and then click **Yes**.

   A message confirms that the table was deleted.

Retrieving a Dropped Table

This section assumes that you created and then dropped the `reg_hist` table, as described in "Dropping a Table" on page 9-28. The following procedure retrieves `reg_hist` from the recycle bin.

To perform the Flashback Drop operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Click **Perform Recovery** in the Backup/Recovery section.

   The Perform Recovery page appears.

3. In the User-Directed Recovery section, select **Tables** from the Recovery Scope list.

   The page reloads with options appropriate for object-level recovery of tables.

4. Select **Flashback Dropped Tables** for the Operation Type, and then click **Recover**.

   The Perform Object Level Recovery: Dropped Objects Selection page appears.

5. Search among the dropped objects in the recycle bin for the objects to retrieve.

   For this example, enter **`hr`** in the **Schema Name** field and **`reg_hist`** in the **Table** field, and click **Go**.

   The Results section lists the objects matching your search. If needed, click the arrow next to Recycle Bin to expand its contents by one level, showing dropped tables matching your search but not their dependent objects.

6. Select each table that you want to retrieve using Flashback Drop.

   For this example, select **`reg_hist`** and then click **Next**.

   > **Note:** When a table is retrieved from the recycle bin, all the dependent objects for the table that are in the recycle bin are retrieved with it. They cannot be retrieved separately.

   The Perform Object Level Recovery: Rename page appears.

7. If needed, enter new names for dropped objects that you are retrieving. For this scenario, do not specify a new name. Click **Next**.

   The primary reason for renaming objects when you retrieve them from the recycle bin is because you might have created new tables with the same names as tables being retrieved. If you need to rename some objects, then enter new names as needed in the **New Name** field.

   The Perform Object Level Recovery: Review page appears. This page displays an impact analysis, showing the full set of objects to be flashed back, including the

dependent objects, as well as the names they will have when the Flashback Drop operation is complete.

8. Review the changes and then click **Submit**.

   A confirmation page should indicate the success of the operation.

9. Click **OK** to return to the Database Home page.

Rewinding a Database Using Oracle Flashback Database

Unlike the other flashback features, Oracle Flashback Database operates at a physical level. When you use Flashback Database, your current datafiles revert to their contents at a previous time. The result is similar to database point-in-time recovery, but Flashback Database can be much faster because it does not require you to restore and recover datafiles. Also, Flashback Database requires limited application of redo data as compared to media recovery.

Flashback Database uses flashback logs to access previous versions of data blocks and also uses some data in the archived redo logs. To have the option of using Flashback Database to repair your database, you must have configured the database to generate flashback logs as explained in "Configuring Recovery Settings" on page 9-6.

To perform a Flashback Database operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Click **Perform Recovery**.

   The Perform Recovery page appears.
3. Complete the following steps:
   a. In the User-Directed Recovery section, select **Whole Database**.
   b. Select **Recover to the current time or a previous point-in-time**.
   c. If necessary, provide the host computer credentials.
   d. Click **Recover**.

   A Confirmation page appears.
4. Click **Yes** to confirm the shutdown of the database.

   The Recovery Wizard page appears. At this point, the shutdown begins.

   When the database is shut down and brought to the mounted state, Database Control is also shut down briefly and restarted. During this process, there is a period during which Database Control cannot respond to your browser. Refresh the page until Database Control responds again.

   When Database Control has restarted and the database is being started and mounted, Database Control may also briefly report that the database is in the `NOMOUNT` state. You are given the choices **Refresh**, **Startup**, and **Perform Recovery**. Refresh the page periodically until the Database Instance page reports that the database instance is mounted before you proceed.
5. Click **Perform Recovery** to resume your recovery session.
6. If you are prompted for host computer and database credentials, then connect with the `SYSDBA` role, or provide host computer credentials for a user in the `DBA` group.

When the Perform Recovery page is displayed again, it shows that the database is mounted, which is required to restore and recover the entire database.

7. Complete the following steps:
   a. In the User-Directed Recovery section, select **Whole Database**.
   b. Select **Recover to the current time or a previous point-in-time**.
   c. If necessary, provide the host credentials.
   d. Click **Recover**.

   A Confirmation page appears.
8. Complete the following steps:
   a. Select **Recover to a prior point-in-time**.
   b. In the **Date** field, select a time 5 minutes before the present time.
   c. Click **Next**.

   The Perform Whole Database Recovery: Flashback page appears.
9. Select **Yes** to specify that you want to use Flashback Database and then click **Next**.

   The Perform Whole Database Recovery: Review page appears.
10. Review the options that you selected and click **Submit**.

    When the flashback operation completes, the Perform Recovery: Result page appears.
11. Click **Open Database**.

    After the database has opened successfully, click **OK**.

Restoring and Recovering the Database

This section demonstrates how to restore and recover the entire database. This example assumes that you are restoring and recovering your database after the loss of one or more datafiles, but you still have a usable server parameter file and control file. You can also use Database Control to restore a lost server parameter file or control file.

To restore and recover the entire database:

1. Perform Step 1 through Step 7 from the procedure in "Rewinding a Database Using Oracle Flashback Database" on page 9-30.
2. Do one of the following:
   - Select **Recover to the current time** to recover all transactions to your database up until the present time.
   - Select **Recover to a prior point-in-time** to recover transactions only up through some previous point in time in the past.

   For this example, select **Recover to the current time**, and then click **Next**.

   The Perform Whole Database Recovery: Rename page appears.
3. Select **No** to restore the files to their default locations, and then click **Next**.

   The Perform Whole Database Recovery: Review page appears.
4. Click **Submit** to start the restoration and recovery of the database.

> **Note:** In some repair scenarios, such as a complete restoration and recovery of your database, the database state will be altered by steps you take while using the wizard. Database Control displays warnings each time a significant database change will result from selecting **Continue** during the recovery process. Pay close attention to these warnings.

When the recovery completes, the Perform Recovery: Result page appears.

5. Click **Open Database** and then **OK**.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for more details about point-in-time recovery

Backup and Recovery: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this section, and includes annotated screenshots. To view the Backup and Recovery OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/backup/backup.htm
```

10

Monitoring and Tuning the Database

Monitoring the performance of a database and ensuring that it performs optimally is an important task for a database administrator. This chapter discusses the features and functions included in Oracle Database that make it easy to monitor database health, identify performance problems, and implement any corrective actions.

This chapter contains the following topics:

- Proactive Database Monitoring
- Diagnosing Performance Problems Using ADDM
- Using Advisors to Optimize Database Performance
- Monitoring and Tuning: Oracle By Example Series

Proactive Database Monitoring

Oracle Database makes it easy to monitor the health and performance of your database. It monitors the vital signs (or metrics) related to database health and performance, analyzes the workload running against the database, and automatically identifies any issues that need your attention as an administrator. The identified issues are presented as alerts and performance findings on the Database Home page. You can also configure Oracle Enterprise Manager Database Control (Database Control) to notify you of issues by e-mail.

This section discusses the following topics:

- About Alerts
- Performance Self-Diagnostics: Automatic Database Diagnostic Monitor
- Monitoring General Database State and Workload
- Managing Alerts

About Alerts

Alerts help you monitor your database. Most alerts notify you of when particular metric thresholds are exceeded. For each alert, you can set critical and warning threshold values. These threshold values are meant to be boundary values that when exceeded, indicate that the system is in an undesirable state. For example, when a tablespace becomes 97 percent full, this can be considered undesirable, and Oracle Database will generate a critical alert.

Other alerts correspond to database events such as Snapshot Too Old or Resumable Session suspended. These types of alerts indicate that the event has occurred.

In addition to notification, you can set alerts to perform some action such as running a script. For instance, scripts that shrink tablespace objects can be useful for a Tablespace Usage warning alert.

By default, Oracle Database issues several alerts, including the following:

- Tablespace Space Used (warning at 85 percent full, critical at 97 percent full)
- Current Open Cursors Count (warning when goes above 1200)
- Session Limit Usage (warning at 90 percent, critical at 97 percent)
- Broken Job Count and Failed Job Count (warning when goes above 0)
- Dump Area Used (warning at 95 percent full)
- Archive Area Used (warning at 80 percent full)

You can modify these alerts and others by setting their metrics.

For more information, see "Managing Alerts" on page 10-7.

Performance Self-Diagnostics: Automatic Database Diagnostic Monitor

Oracle Database includes a self-diagnostic engine called Automatic Database Diagnostic Monitor (ADDM). ADDM makes it possible for Oracle Database to diagnose its own performance and determine how any identified problems can be resolved.

To facilitate automatic performance diagnosis using ADDM, Oracle Database periodically collects snapshots of the database state and workload. **Snapshots** are sets of historical data for specific time periods that are used for performance comparisons by ADDM. The default collection interval for a snapshot is one hour. Snapshots provide a statistical summary of the state of the system at a point in time. These snapshots are stored in Automatic Workload Repository (AWR), residing in the `SYSAUX` tablespace. The snapshots are stored in this repository for a set time (8 days by default) before they are purged to make room for new snapshots.

ADDM analyzes data captured in AWR to determine the major problems in the system, and in many cases, recommends solutions and quantifies expected benefits. ADDM analysis results are represented as a set of **findings**.

Generally, the performance problems that ADDM can call attention to include the following:

- Resource contention (bottlenecks), such as when your database is using large amounts of CPU time or memory due to high load SQL statements
- Poor connection management, such as when your application is making too many logins to the database
- Lock contention in a multiuser environment, such as when one user process acquires a lock to safely update data in a table, causing other user processes that need to acquire locks against the same table to wait, resulting in a slower database performance

See Also:

- *Oracle Database Performance Tuning Guide*

Monitoring General Database State and Workload

The Database Home page (Figure 10–1) enables you to monitor the state and workload of your database. It provides a central place for general database state information and is updated periodically.

Figure 10–1 Database Home Page

To monitor the general database state and workload:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. (Optional) Click the **Refresh** button to update the information displayed.

   By default, the Database Home page automatically refreshes every 60 seconds. You can prevent automatic refresh by selecting **Manually** in the View Data list at the top right-hand corner of the page. You must then click **Refresh** to view the latest information.

   The date and time that data was last collected from the database is displayed to the left of the Refresh button.

3. Get a quick overview of the database state in the General section, which includes the following information:

   - Status of the database instance, Up or Down

     Click the **Status** link to drill down to database availability details.

- Time the database was last started
- Instance name
- Oracle Database version
- Host name

  Click the **Host** link to drill down to host details.

- Listener name

  Click the **Listener** link to drill down to listener details.

Click **View All Properties** to see the Oracle home path and whether the database is read-only or read/write.

4. View CPU utilization in the Host CPU section, which includes the following information:

   - Bar chart

     This chart shows the percentage of CPU time used by the database and other processes. The chart legend contains links for the database instance and for other CPU processes.

     Click the **Other** link in the chart legend to see how the utilization of CPU, memory, and disk I/O change over time.

     Click the instance name link in the chart legend to see the Top Activity page. It includes a graph of active sessions over time, details about SQL statements issued, and the most active sessions.

   - CPU load

     This is the average number of processes waiting to be scheduled for the CPU in the previous minute.

     Click the **Load** link to see how the utilization of CPU, memory, and disk I/O change over time.

   - Paging

     This is the number of memory pages (fixed-length block of instructions, data, or both) that are paged out (moved out of active memory) each second.

     Click the **Paging** link to see how the utilization of CPU, memory, and disk I/O change over time.

5. Investigate the Active Sessions section, where you can further explore the cause of performance problems, such as your database taking up most of the CPU time on the server. This section displays a bar graph with the following information:

   - Waits

     This is the value for all wait classes combined, excluding user I/O and idle wait events. **Wait classes** are groupings of wait events based on the type of wait.

     If other processes are taking up most of your CPU time, then this indicates that some other application running on the database host computer could be causing performance problems.

     Click the **Wait** link to go to the Performance page to view potential problems inside and outside the database.

   - User I/O

This is the average number of active sessions waiting for user I/O. User I/O means that the workload originating from the user causes the database to read data from disk or write data to disk.

Click the **User I/O** link to go to the Performance page to view potential problems inside and outside the database.

- CPU

  This is the average active sessions using CPU.

  Click the **CPU** link to see a chart showing more detailed information about active sessions over time.

6. View the Diagnostic Summary section, which includes the following information:

   - ADDM Findings

     This shows the count of ADDM findings from the most recent ADDM run. Click the number adjacent to the ADDM Findings link to go to the ADDM page.

   - Period Start Time

     This is the start time of the time period most recently analyzed by ADDM. It is shown only if there are ADDM findings.

   - Alert Log

     This is the timestamp of the most recent alert log entry that describes an `ORA-` error.

     Click the **Alert Log** link to go to the Alert Log Errors page, which shows a list of log entries that contain errors.

   - Active Incidents

     This shows the count of active *incidents*, which are occurrences of critical errors in the database. You are encouraged to investigate critical errors and report them to Oracle Support Services. Click the count to go to the Support Workbench home page.

   - Database Instance Health

     Click **Database Instance Health** to display the Database Instance Health page, which includes graphical timelines of incidents, ADDM findings, and alerts. You can use these graphs for identifying correlations between incidents and alerts generated and performance issues on the system.

7. View the SQL Response Time section.

   This is the current response time of a tracked set of SQL statements as compared to the response time for the reference collection. A **reference collection**, or SQL Tuning Set, is set of SQL statements that represents the typical SQL workload on your production system. If the current response time and reference collection response time are equal, then the system is running as it should. If the current response time is greater than the reference collection response time, then one or more SQL statements are performing more slowly than they should. The lower the current response time, the more efficiently the tracked SQL statements are running.

   Click the **SQL Response Time** link to see response time metrics for the previous 24 hours. If the reference collection is empty, then click **Reset Reference Collection** to go to a page where you can create a reference collection.

8. View the Space Summary section.

   If the number adjacent to the Segment Advisor Recommendations label is not zero, it means the Segment Advisor has identified candidate segments for space defragmentation. Click the number to view recommendations for how to defragment these segments.

9. View the Alerts section, which includes the following items:

   - Category list

     Optionally choose a category from the list to view alerts only in that category.

   - Critical

     This is the number of metrics that have exceeded critical thresholds plus the number of other critical alerts, such as those caused by incidents (critical errors).

   - Warning

     This is the number of metrics that have exceeded warning thresholds.

   - Alerts table

     Click the message to learn more about the alert.

10. View the ADDM Performance Analysis section, if present. This section contains the following items:

    - Period Start Time

      This is the start time of the period most recently analyzed by ADDM.

    - Period Duration in minutes

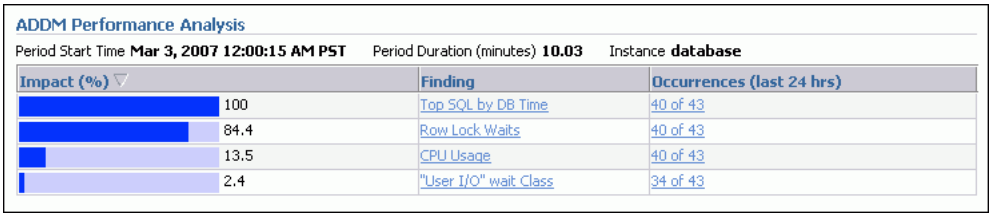
      This is the duration of the period most recently analyzed by ADDM.

    - Instance name

    - ADDM findings table

      This table lists the ADDM findings, their estimated impact on database performance, a description of the finding, and the number of times the finding occurred in snapshots collected during the previous 24 hours. For example, a finding with Occurrences listed as 34 of 43 has occurred in 34 of the last 43 snapshots.

    Click the finding to view finding details, to view recommendations, and in some cases to implement recommendations or start advisors.



To view database performance over time:

1. At the top of the Database Home page, click **Performance**.

   The Performance page appears, displaying a summary of CPU utilization, average active sessions, instance disk I/O, and instance throughput for the recent time period.

2. Use the Additional monitoring links to drill down to Top Activity and other data.

The types of actions you can take to improve host performance depends on your system, and can include eliminating unnecessary processes, adding memory, or adding CPUs.

Managing Alerts

The following topics describe how to manage alerts:

- Viewing Metrics and Thresholds
- Setting Metric Thresholds
- About Responding to Alerts
- Clearing Alerts
- Setting Up Direct Alert Notification

Viewing Metrics and Thresholds

To effectively diagnose performance problems, statistics must be available. Oracle generates many types of cumulative statistics for the system, sessions, and individual SQL statements. Oracle also tracks cumulative statistics on segments and services. A **metric** is defined as the rate of change in some cumulative statistic. Metrics are computed and stored in Automatic Workload Repository, and are displayed on the All Metrics page, which can be viewed by clicking **All Metrics** under Related Links on the Database Home page.

To view metrics for your database:

1. On the Database Home page under Related Links, click **All Metrics**.

   The All Metrics page appears.

2. Click a specific metric link.

   A details page appears, with more information about the metric. Online Help for this page describes the metric.

All Metrics

Collected From Target **Feb 23, 2007 10:58:34 AM PST**

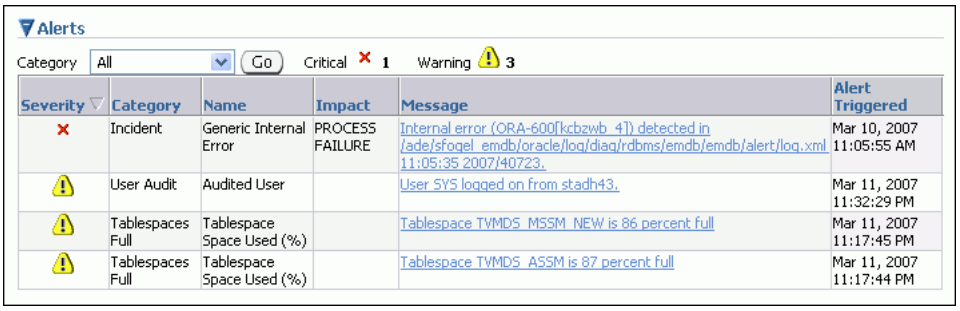
Expand All | Collapse All

Metrics	Thresholds	Collection Schedule	Upload Interval	Last Upload
orcl.us.oracle.com				
Alert Log	Some	Every 15 Minutes	Every Collection	Feb 7, 2007 3:49:15 PM
Alert Log Content	None	Real-time Only	n/a	n/a
Alert Log Error Status	All	Every 15 Minutes	Every Collection	Feb 23, 2007 10:49:15 AM
Archive Area	Some	Every 15 Minutes	Every Collection	Feb 23, 2007 10:46:17 AM
Database Files	None	Every 10 Minutes	Every Collection	Server Generated
Database Job Status	All	Every 5 Minutes	On Alert	Feb 3, 2007 2:05:50 AM
Database Limits	Some	Every 5 Minutes	Every Collection	Server Generated
Database Services	None	Every 15 Minutes	Every Collection	Server Generated
Deferred Transactions	All	Every 5 Minutes	On Alert	Jan 27, 2007 1:26:50 AM
Dump Area	Some	Every 15 Minutes	Every Collection	Feb 23, 2007 10:46:17 AM
Efficiency	None	Every 5 Minutes	Every Collection	Server Generated
Failed Logins	All	Every 30 Minutes	Every Collection	Feb 20, 2007 2:33:52 PM
Flash Recovery	None	Every 5 Minutes	Every Collection	Feb 23, 2007 10:55:50 AM
Health Check	Some	Every 15 Seconds	On Alert	Feb 3, 2007 2:04:04 AM
Incident	Some	Every 5 Minutes	Every Collection	Feb 7, 2007 3:45:23 PM
Incident Status	All	Every 5 Minutes	Every Collection	Feb 23, 2007 10:55:23 AM

For each of these metrics, you are able to define warning and critical threshold values, and whenever the threshold is exceeded, Oracle Database issues an alert. Alerts are displayed on the Database Home page under the Alerts heading (or Related Alerts for nondatabase alerts such as a component of Oracle Net).

Figure 10–2 shows two warning alerts for the threshold Tablespace Space Used.

Figure 10–2 Alerts Section of Database Home Page



Setting thresholds is discussed in "Setting Metric Thresholds" on page 10-8. Actions you might take to respond to alerts are discussed in "About Responding to Alerts" on page 10-10.

When the condition that triggered the alert is resolved and the metric value is no longer outside the boundary, Oracle Database clears the alert. Metrics are important for measuring the health of the database and serve as input for self-tuning and recommendations made by Oracle Database advisors.

Setting Metric Thresholds

Oracle Database provides a set of predefined metrics, some of which have predefined thresholds. There may be times when you want to set thresholds for other metrics, or you want to alter existing threshold settings.

One means of setting a threshold is described in "Changing Space Usage Alert Thresholds for a Tablespace" on page 6-16, where you set warning and critical thresholds on the amount of space consumed in a tablespace. A more general means of setting thresholds is available using the Edit Thresholds page.

To set metric thresholds:

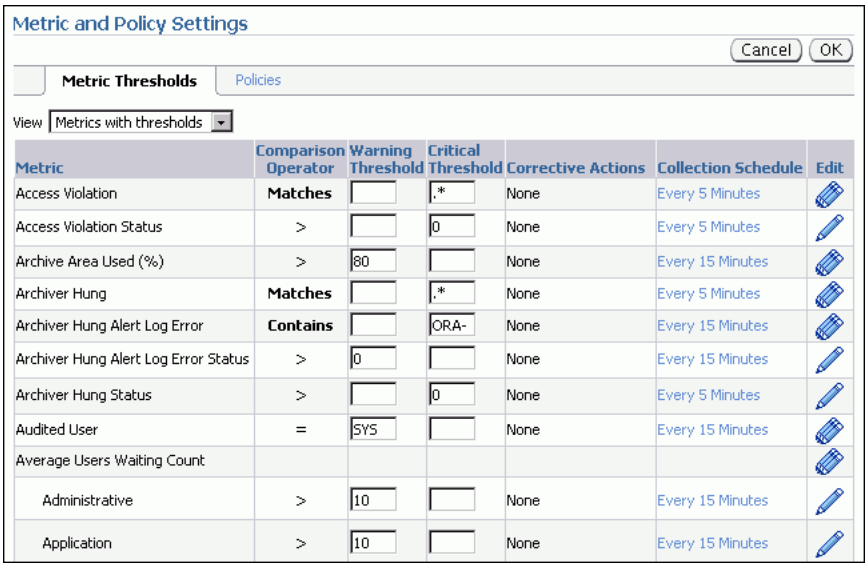
1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. Under the Related Links heading, click **Metric and Policy Settings**.

   The Metric and Policy Settings page appears.

   This page displays the existing thresholds for metrics and any response actions that have been specified.



3. In the View list, do one of the following:
   - Select **Metrics with thresholds** to view only those metrics with thresholds, either predefined by Oracle or previously set by you.
   - Select **All Metrics** to view all metrics, whether or not they have thresholds defined.
4. To set or modify a warning threshold for a particular metric, enter the value you want in the **Warning Threshold** field for that metric.
5. To set or modify a critical threshold for a particular metric, enter the value you want in the **Critical Threshold** field for that metric.
6. To disable or reenable metric collection for a particular metric, or to change its collection schedule, complete the following steps:
   a. Click the Collection Schedule link for the metric.

      The Edit Collection Settings page for that metric appears.
   b. Click **Disable** to disable collection for this metric, or click **Enable** to enable it.
   c. Choose the scale for your collection schedule from the Frequency Type list.
   d. Enter a number in the **Repeat Every** field.
   e. Do one of the following:
      - Click **Continue** to save your choices and return to the Metric and Policy Settings page.
      - Click **Cancel** to return to the Metric and Policy Settings page without saving your choices.
7. Click a single-pencil icon to use the Edit Advanced Settings page to make changes to Corrective Actions, (Monitoring) Template Override, and Advanced Threshold Settings.
8. Click a triple-pencil icon to set different threshold values for different instances of the object type being measured.

   For example, for each tablespace you can set different warning and critical levels for the Tablespace Space Used metric.

9. Do one of the following:
   - Click **OK** to save your changes and return to the Database Home page.
   - Click **Cancel** to return to the Database Home page without saving your changes.

About Responding to Alerts

When you receive an alert, follow any recommendations it provides, or consider running ADDM or another advisor, as appropriate, to get more detailed diagnostics of system or object activity.

For example, if you receive a Tablespace Space Usage alert, you might take a corrective measure by running the Segment Advisor on the tablespace to identify possible objects for shrinking. You can then shrink the objects to create available (free) space. See "Reclaiming Unused Space" on page 6-19.

Additionally, as a response, you can set a corrective script to run as described in "Setting Metric Thresholds" on page 10-8.

Clearing Alerts

Most alerts are cleared (removed) automatically when the cause of the problem disappears. Other alerts, such as Generic Alert Log Error, are sent to you for notification and need to be acknowledged by the database administrator.

After taking the necessary corrective measures, you can acknowledge an alert by clearing or purging it. Clearing an alert sends the alert to the Alert History, which can be viewed from the Database Home page under Related Links. Purging an alert removes it from the Alert History.

To clear or purge an alert:

1. On the Database Home page under Diagnostic Summary, click the **Alert Log** link.

   The Alert Log Errors page appears.
2. From the View Data list, select the period for which you want information.
3. Click **Refresh** to refresh the page with the latest information.
4. Do one of the following:
   - Click **Show Open Alerts** to hide alerts that have been cleared.
   - Click **Show Open and Cleared Alerts** to see all alerts.

   Note: You will see only one or the other of these buttons, depending on what is currently displayed.
5. Select one or more alerts by clicking their **Select** options.
6. Click **Clear** to clear the alert.
7. Click **Purge** to purge the alert.
8. Click **Clear Every Open Alert** to clear all open alerts.
9. Click **Purge Every Alert** to purge all alerts.

Setting Up Direct Alert Notification

Database Control displays all alerts on the Database Home page. However, you can optionally specify that Database Control provide you direct notification when specific alerts arise. For example, if you specify that you want e-mail notification for critical alerts, and you have a critical threshold set for the *system response time for each call* metric, then you might be sent an e-mail message similar to the following:

```
Host Name=mydb.us.mycompany.com
Metric=Response Time per Call
Timestamp=08-NOV-2006 10:10:01 (GMT -7:00)
Severity=Critical
Message=Response time per call has exceeded the threshold. See the
latest ADDM analysis.
Rule Name=
Rule Owner=SYSMAN
```

The e-mail message contains a link to the host name and the latest ADDM analysis.

By default, alerts in critical state such as DB Down, Generic Alert Log Error Status, and Tablespace Space Used are set up for notification. However, to receive these notifications, you must set up your e-mail information.

To set up your e-mail information:

1. From any Database Control page, click the **Setup** link, which is visible in the header and footer areas.
2. On the Setup page, select **Notification Methods**.
3. Enter the required information into the Mail Server section of the Notifications Methods page. Click **Help** at the bottom of the page for assistance.

   There are other methods of notification, including scripts and SNMP (Simplified Network Management Protocol) program interrupts (traps). The latter can be used to communicate with third-party applications.

   At this point, you have set up a method of notification, but you have not set up an e-mail address to receive the notification. To do so, complete the following steps.
4. From any Database Control page, click the **Preferences** link, which is visible in the header and footer areas.
5. On the Preferences page, select **General**. Click **Add Another Row** in the E-mail Addresses section to enter your e-mail address.
6. Click **Test** to verify that e-mail messages can be sent using the specified information. After the test completes, click **OK**.
7. (Optional) To edit notification rules, such as to change the severity state for receiving notification, select **Rules** under the heading Notification on the left-hand side of the page.

   The Notification Rules page appears. Click **Help** for more information about this page.

See Also:

- *Oracle Enterprise Manager Advanced Configuration*,for more information about configuring notification rules
- "Viewing Metrics and Thresholds"
- "Setting Metric Thresholds"

Diagnosing Performance Problems Using ADDM

At times, database performance problems arise that require your diagnosis and correction. In most cases, these problems are brought to your attention by Automatic Database Diagnostic Monitor (ADDM), which does a top-down system analysis every hour by default, and reports its most significant findings on the Database Home page.

This section contains the following topics:

- Viewing a Summary of ADDM Performance Findings
- Responding to ADDM Performance Findings
- Running ADDM Manually
- Modifying AWR Snapshot Frequency and Retention

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*.

Viewing a Summary of ADDM Performance Findings

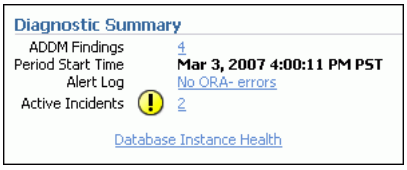
ADDM analysis results consist of a description of each finding and a recommended action. You can view a summary of findings and their impacts on the system.

To view a summary of ADDM performance findings:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

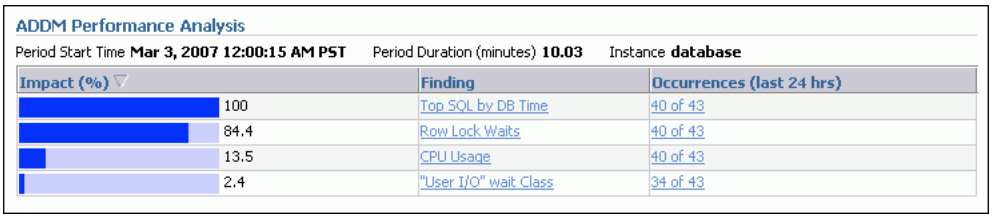
2. In the Diagnostic Summary section of the Database Home page, locate the numeric link next to the ADDM Findings label.



   The number indicates the number of findings from the most recent ADDM run. If this number is zero, there are no ADDM findings to investigate.

   If this number is not zero, continue to Step 3.

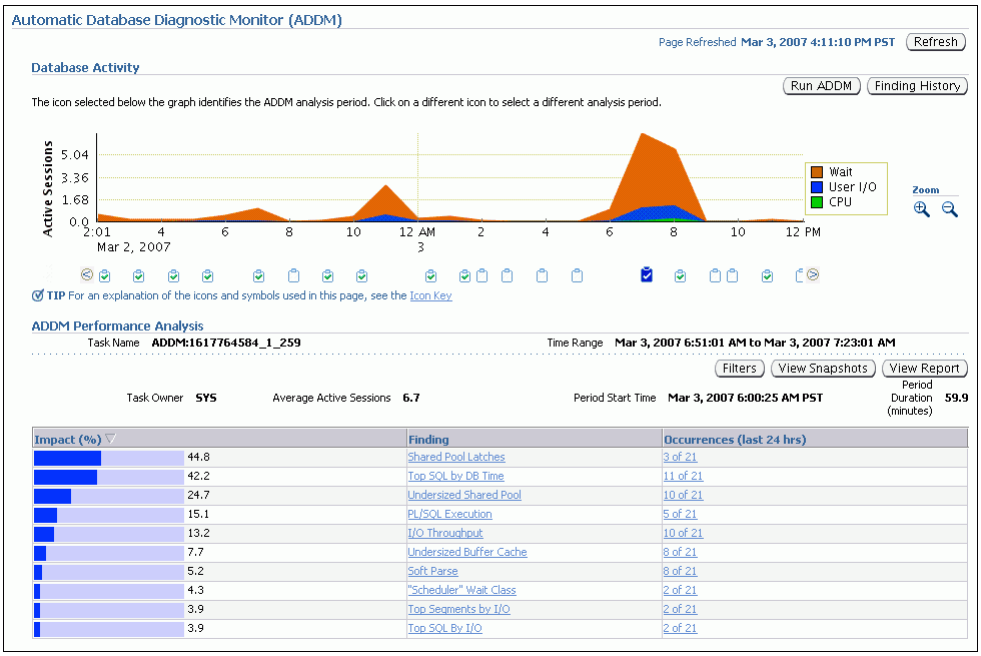
3. View the ADDM Performance Analysis section, below the Alerts section of the Database Home page.



   This section is present only if there are ADDM findings. It displays the first five findings from the most recent ADDM run. If there are more than five findings, controls appear to enable you to view the next five findings, and so on.

4. In the Diagnostic Summary section of the Database Home page, locate the nonzero numeric link next to the ADDM Findings label, and click it.

The ADDM page appears, showing a summary of all findings from the most recent run, and a graphical timeline showing session activity over the past 24 hours.



Click **Help** for more information about this page.

Each clipboard icon beneath the graph represents a time range within the 24 hours. Click an icon to view findings for the ADDM run that took place during that icon's time range. If more than one ADDM run occurred during that time range, you can select an individual ADDM run from the list that appears next to the Task Name label.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*.

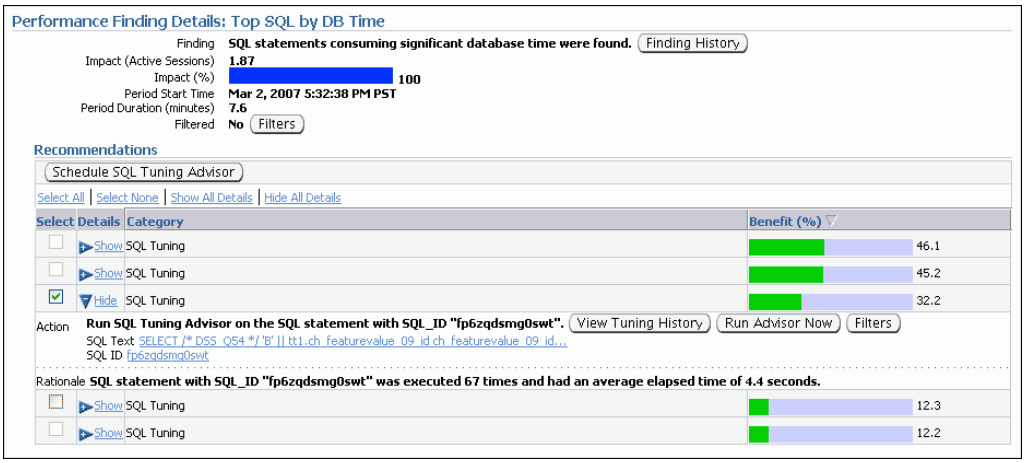
Responding to ADDM Performance Findings

You can act upon the recommendations that accompany ADDM performance findings.

To respond to ADDM performance findings:

1. In the ADDM Performance Analysis section of either the Database Home page or the ADDM page, in the **Finding** column, click a finding.

   The Performance Finding Details page appears.



2. In the Recommendations section, click **Show All Details**.
3. Choose a finding to respond to, and follow the recommended action. A recommendation can include running an advisor, which you can do by clicking **Run Advisor Now**.
4. (Optional) Select one or more findings, and then click the action button above the table of findings. An example of an action button might be Schedule SQL Tuning Advisor.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

Running ADDM Manually

By default, Oracle Database runs ADDM every hour, immediately after an AWR snapshot. Performance findings from the most recent ADDM run, if any, are listed on the Database Home page. This is described in "Performance Self-Diagnostics: Automatic Database Diagnostic Monitor" on page 10-2.

You can also run ADDM manually. Reasons for doing so include running it as a recommended action associated with an alert or running it across multiple snapshots.

Note: If you need more frequent ADDM reporting, you can also modify the default snapshot interval. To do so, see "Modifying AWR Snapshot Frequency and Retention" on page 10-16.

The following steps describe how to run ADDM to analyze a period that spans multiple snapshots.

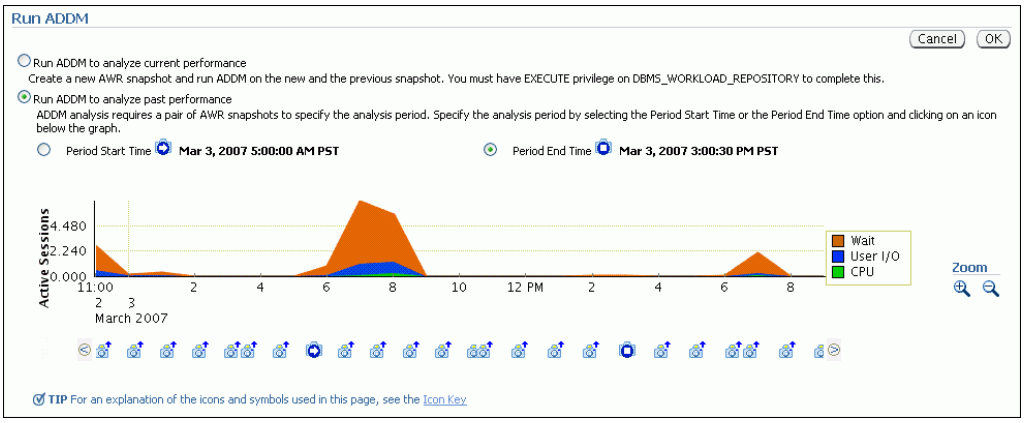
To run ADDM manually:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.
2. Under Related Links at the bottom of the page, click **Advisor Central**.

   The Advisor Central page appears.
3. Click **ADDM**.

   The Run ADDM page appears.



4. Select **Run ADDM to analyze past performance**.
5. Complete the following steps to choose a period start time:
   a. Select the **Period Start Time** option.
   b. (Optional) Examine the timeline graph and note the number of snapshots available in each 2-hour period. Each snapshot is represented by a camera icon. If your snapshots occur more frequently than once each hour and you want to view the timeline of snapshots with more precision, click the zoom-in icon at the right. (The zoom-in icon is a magnifying glass with a plus-sign.)
   c. (Optional) Below the timeline graph, click the left-arrow icon or right-arrow icon at the left and right sides of the page to change the period shown in the timeline graph.
   d. Click a camera icon.

      A bold right arrow indicating period start time covers the camera icon, and the Period Start Time field above the timeline graph changes to the selected date and time.
6. If you want a period end time other than the default (the most recent snapshot), complete the following steps:
   a. Above the timeline graph, select the **Period End Time** option.
   b. (Optional) Click the left-arrow icon or right-arrow icon at the left and right sides of the page to change the period shown in the timeline graph.
   c. Click a camera icon.

      A bold square (like a stop button on a recording device) covers the camera icon, and the Period End Time field changes to the selected date and time.
7. Click **OK** to start the analysis.

   After a short delay, the Automatic Database Diagnostic Monitor page appears, showing its findings.
8. (Optional) In the Task Name list, note the task name.

   A typical task name is "Task_52." You can return to this analysis at a later time by searching for and selecting the task name in the Advisor Central page.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*.

Modifying AWR Snapshot Frequency and Retention

ADDM analysis is based on AWR snapshots, which have a default frequency of once each hour and a default retention period of 8 days.

> **Note:** The snapshot retention period does not apply to preserved snapshots, which are retained forever. A preserved snapshot set is typically created to define a reference period for performance analysis.

You can view and alter these AWR snapshot settings.

To modify AWR snapshot frequency and retention:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the top of the page, click **Server** to display the Server page.
3. In the Statistics Management section, click **Automatic Workload Repository**.

   The Automatic Workload Repository page appears.

4. Click **Edit**.

   The Edit Settings page appears.

5. Enter new snapshot retention settings or new snapshot collections settings, and then click **OK**.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*.

Using Advisors to Optimize Database Performance

Oracle Database includes a set of *advisors* to help you manage and tune your database. This section contains background information about these advisors and instructions for their use. The following topics are covered:

- About Advisors
- About the SQL Advisors
- About the Automatic SQL Tuning Advisor
- Configuring the Automatic SQL Tuning Advisor
- Viewing Automatic SQL Tuning Results
- Running the SQL Tuning Advisor
- Running the SQL Access Advisor
- Using the Memory Advisors

About Advisors

Advisors are powerful tools for database management. They provide specific advice on how to address key database management challenges, covering a wide range of areas including space, performance, and undo management. In general, advisors produce more comprehensive recommendations than alerts. This is because alert generation is intended to be low cost and have minimal impact on performance,

whereas advisors consume more resources and perform more detailed analysis. This, along with the what-if capability of some advisors, provides vital information for tuning that cannot be procured from any other source. You run advisors from Oracle Enterprise Manager Database Control (Database Control). Some are also run automatically during maintenance windows (time periods).

Advisors are provided to help you improve database performance. These advisors include Automatic Database Diagnostic Monitor (ADDM), SQL advisors, and memory advisors. For example, one of the memory advisors, the Shared Pool Advisor, graphically displays the impact on performance of changing the size of this component of the System Global Area (SGA).

You can run a performance advisor when faced with the following situations:

- You want to resolve a problem in a specific area, for example, to determine why a given SQL statement is consuming 50 percent of CPU time and what to do to reduce its resource consumption.You can use the SQL Tuning Advisor.
- During application development, you want to tune a new schema and its associated SQL workload for optimal performance. You can use the SQL Access Advisor.
- You are planning to add memory to your system. You can use the Memory Advisor to determine the database performance impact of increasing your SGA or PGA (Program Global Area).

You can run all advisors from the Advisor Central home page, accessible through a link on the Database Home page. You can also invoke many of the advisors from the Performance page, through recommendations from ADDM, or from alerts.

Table 10–1 describes the performance advisors. Other advisors are listed in Table 10–2.

Table 10–1 Performance Advisors

Advisor	Description
Automatic Database Diagnostic Monitor (ADDM)	ADDM makes it possible for Oracle Database to diagnose its own performance and determine how any identified problems can be resolved. See "Performance Self-Diagnostics: Automatic Database Diagnostic Monitor" on page 10-2 and "Diagnosing Performance Problems Using ADDM" on page 10-12.
SQL Advisors ■ SQL Tuning Advisor ■ SQL Access Advisor	The SQL Tuning Advisor analyzes one or more SQL statements and makes recommendations for improving performance. This advisor is run automatically during the maintenance periods, but can also be run manually. See "About the Automatic SQL Tuning Advisor" on page 10-19 and "Running the SQL Tuning Advisor" on page 10-24. The SQL Access Advisor tunes a schema to a given SQL workload. For example, the SQL Access Advisor can provide recommendations for creating indexes, materialized views, or partitioned tables for a given workload. See "Running the SQL Access Advisor" on page 10-26.

Table 10–1 (Cont.) Performance Advisors

Advisor	Description
Memory Advisors ■ Memory Advisor ■ SGA Advisor ■ Shared Pool Advisor ■ Buffer Cache Advisor ■ PGA Advisor	The Memory Advisors provide graphical analyses of total memory target settings, SGA and PGA target settings, or SGA component size settings. You use these analyses to tune database performance and for what-if planning. Depending on the current memory management mode, different memory advisors are available. ■ If Automatic Memory Management is enabled, only the Memory Advisor is available. This advisor provides advice for the total memory target for the instance. ■ If Automatic Shared Memory Management is enabled, the SGA Advisor and PGA Advisor are available. ■ If Manual Shared Memory Management is enabled, the Shared Pool Advisor, Buffer Cache Advisor, and PGA Advisor are available. See "Using the Memory Advisors" on page 10-27. See "Managing Memory" on page 5-12 for more information about memory management modes.

Table 10–2 Other Advisors

Advisor	Description
Segment Advisor	The Segment Advisor provides advice on whether or not a segment is a good candidate for a shrink operation based on the level of space fragmentation within that segment. The advisor also reports on the historical growth trend of segments. You can use this information for capacity planning and for arriving at an informed decision about which segments to shrink. See "Reclaiming Unused Space" on page 6-19.
Undo Advisor	The Undo Advisor assists in correctly sizing the undo tablespace. The Undo Advisor can also be used to set the low threshold value of the undo retention period for any Oracle Flashback requirements. See "Computing the Minimum Undo Tablespace Size Using the Undo Advisor" on page 6-28.

About the SQL Advisors

The SQL advisors examine a given SQL statement or a set of SQL statements and provide recommendations to improve efficiency. These advisors can make various types of recommendations, such as creating **SQL profiles** (a collection of information that enables the query optimizer to create an optimal execution plan for a SQL statement), restructuring SQL statements, creating additional indexes, materialized views, or partitions, and refreshing optimizer statistics. Oracle Enterprise Manager Database Control (Database Control) enables you to accept and implement many of these recommendations with just a few mouse clicks.

The two SQL advisors are the SQL Tuning Advisor and the SQL Access Advisor.

SQL Tuning Advisor

You use the SQL Tuning Advisor to tune a single or multiple SQL statements. Typically, you run this advisor in response to an ADDM performance finding that recommends its use. You can also run it periodically on the most resource-intensive SQL statements, as well as on a SQL workload.

When tuning multiple SQL statements, the SQL Tuning Advisor does not recognize interdependencies between the SQL statements. It solves SQL performance problems

by identifying problems with individual SQL statements, such as a poorly performing optimizer plan or the mistaken use of certain SQL structures.

You can run the SQL Tuning Advisor against the following sources:

- Top Activity—The most resource-intensive SQL statements executed during the last hour. Use this option to tune SQL statements that might have caused recent performance problems.
- Historical SQL— A set of SQL statements over any 24 hour window (time period). Use this option for proactive tuning of SQL statements.
- SQL Tuning Sets (STS)—A set of SQL statements you provide. An STS can be created from SQL statements captured by AWR snapshots or from any SQL workload.

SQL Access Advisor

The SQL Access Advisor is primarily responsible for making schema modification recommendations. It can recommend that you create access structures such as indexes and materialized views to optimize SQL queries. It can also recommend that you partition tables, indexes, or materialized views to improve query performance.

The SQL Access Advisor takes a SQL workload as input. You can select your workload from different sources, including current and recent SQL activity, a SQL repository, or a user-defined workload such as from a development environment. The advisor then makes recommendation to improve the performance of the workload as a whole.

Statement Tuning and Workload Tuning

Note that both the SQL Tuning Advisor and the SQL Access Advisor provide index creation recommendations. The SQL Tuning Advisor recommends creation of indexes only when it anticipates exceptional performance gains for the SQL statement being tuned. However, creation of new indexes may adversely impact the performance of DML insert, update, and delete operations. The SQL Tuning advisor does not take this into account while generating new index recommendations.

The SQL Access Advisor, on the other hand, considers the impact of new indexes on the complete workload. As such, if an index improves performance of one SQL statement but adversely impacts the rest of the workload, the new index will not be recommended by the SQL Access Advisor. For this reason, the SQL Tuning Advisor always recommends validating its new index recommendation by running the SQL Access Advisor.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide* for more information about tuning SQL statements with the SQL Tuning Advisor
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about the SQL Access Advisor
- "Running the SQL Tuning Advisor" on page 10-24
- "Running the SQL Access Advisor" on page 10-26

About the Automatic SQL Tuning Advisor

Beginning with Oracle Database 11*g*, the SQL Tuning Advisor runs automatically during system maintenance windows (time periods) as a maintenance task. During

each automatic run, the advisor selects high-load SQL queries in the system and generates recommendations on how to tune these queries.

The Automatic SQL Tuning Advisor can be configured to automatically implement SQL profile recommendations. A **SQL profile** contains additional SQL statistics that are specific to the SQL statement and enable the query optimizer to generate a significantly better execution plan at run time. If you enable automatic implementation, then the advisor creates SQL profiles for only those SQL statements where the performance increase would be at least threefold. Other types of recommendations, such as the creation of new indexes, refreshing optimizer statistics, or restructuring SQL, can be implemented only manually. DML statements are not considered for tuning by the Automatic SQL Tuning Advisor.

You can view a summary of the results of automatic SQL tuning over a specified period (such as the previous 7 days), and a detailed report about recommendations made for all SQL statements that the SQL Tuning Advisor has processed. You can then implement selected recommendations. You can also view the recommendations that were automatically implemented.

You can control when the Automatic SQL Tuning Advisor runs, and you can disable it altogether if desired.

See Also:

- "Viewing Automatic SQL Tuning Results" on page 10-23
- "Configuring the Automatic SQL Tuning Advisor" on page 10-20
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database Administrator's Guide* for more information about automated maintenance tasks

Configuring the Automatic SQL Tuning Advisor

The following are some of the configuration tasks that you might want to perform for the Automatic SQL Tuning Advisor:

- Enable automatic implementation of SQL profile recommendations.

  Automatic implementation is disabled by default.
- Select the maintenance windows (time periods) in which the advisor runs.

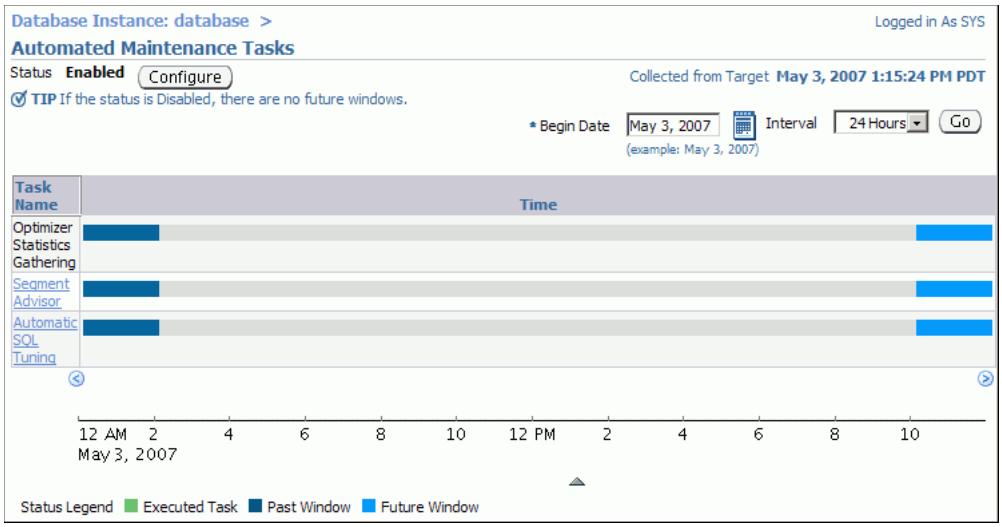
  The Automatic SQL Tuning Advisor runs in all maintenance windows by default.
- Modify the start time and duration of existing maintenance windows or create new maintenance windows.

To configure the Automatic SQL Tuning Advisor:

1. Go to the Database Home page, logging in as user `SYS`.

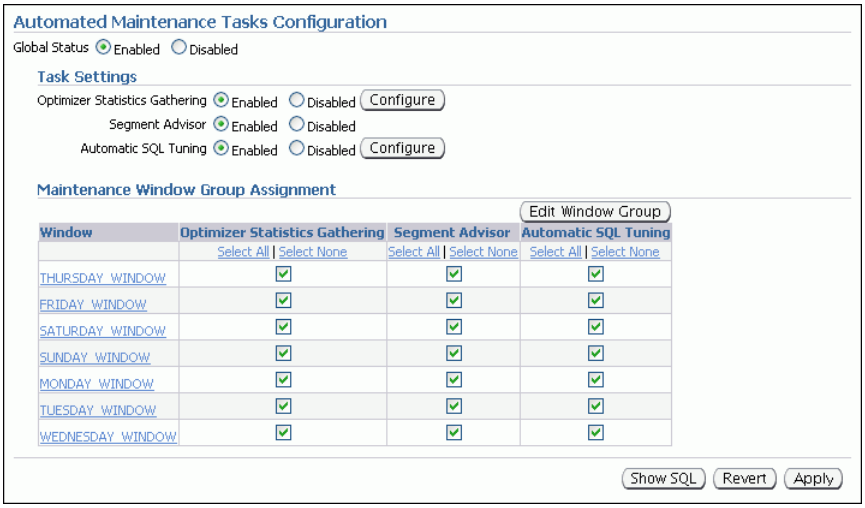
   See "Accessing the Database Home Page" on page 3-4.
2. Click **Server** to display the Server page.
3. In the Oracle Scheduler section, click **Automated Maintenance Tasks**.

   The Automated Maintenance Tasks page appears.



4. Click **Configure**.

   The Automated Maintenance Tasks Configuration page appears.

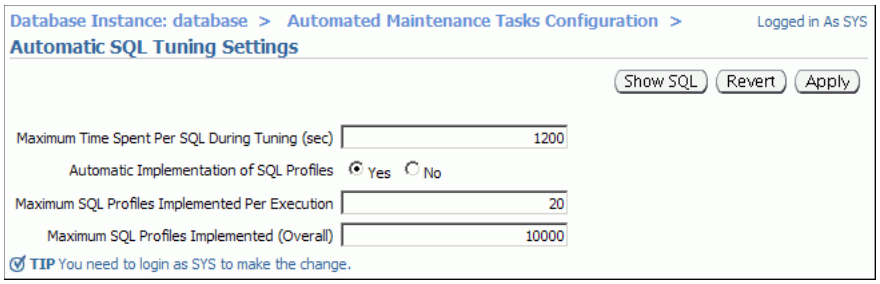


5. (Optional) To disable the Automatic SQL Tuning Advisor altogether, in the Task Settings section, click the **Disabled** option for Automatic SQL Tuning.
6. (Optional) To prevent the Automatic SQL Tuning Advisor from running in particular maintenance windows, in the Maintenance Window Group Assignment section, deselect check boxes under the Automatic SQL Tuning heading.
7. Click **Apply** to save any changes made so far.

   A confirmation message is displayed.
8. To enable automatic implementation of SQL profile recommendations, complete the following steps:

   a. In the Task Settings section, next to the Automatic SQL Tuning options, click **Configure**.

      The Automatic SQL Tuning Settings page appears.



b. Next to Automatic Implementation of SQL Profiles, click the **Yes** option.

You must be logged in as user `SYS` to change this option.

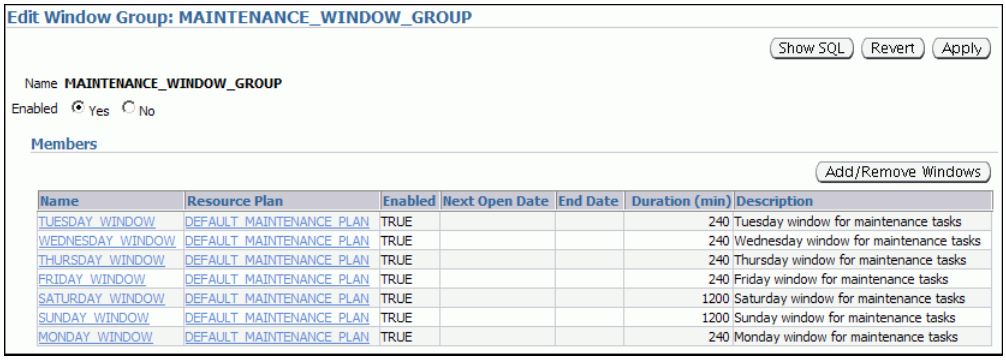
c. Click **Apply**.

A confirmation message is displayed.

d. In the *locator links* at the top, left-hand side of the page, click **Automated Maintenance Tasks Configuration** to return to the Automated Maintenance Tasks Configuration page.

9. (Optional) To make changes to the start time and duration of existing maintenance windows, to disable individual maintenance windows, or to create additional maintenance windows, click **Edit Window Group**.

The Edit Window Group page appears. From this page you can change the settings of individual windows or you can add or remove windows to or from the window group `MAINTENANCE_WINDOW_GROUP`.



See the online Help for this page for more information.

Note: If you create a new window (time period) to run automated maintenance tasks, you must add that window to `MAINTENANCE_WINDOW_GROUP`.

See Also:

- *Oracle Database Administrator's Guide* for more information about maintenance windows and how to configure them
- "About the Automatic SQL Tuning Advisor" on page 10-19

Viewing Automatic SQL Tuning Results

You can track the activities of the Automatic SQL Tuning Advisor with Database Control.

To view automatic SQL tuning results:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

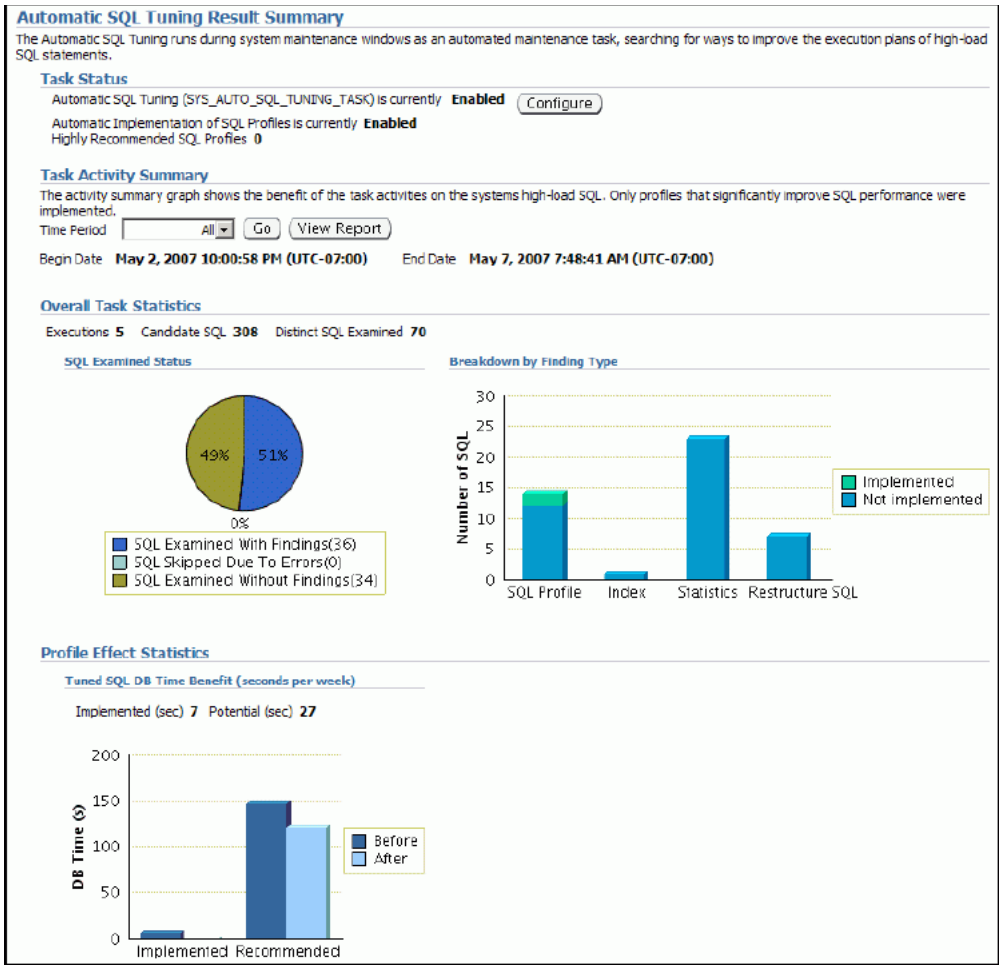
2. Click **Server** to display the Server page.

3. In the Oracle Scheduler section, click **Automated Maintenance Tasks**.

   The Automated Maintenance Tasks page appears.

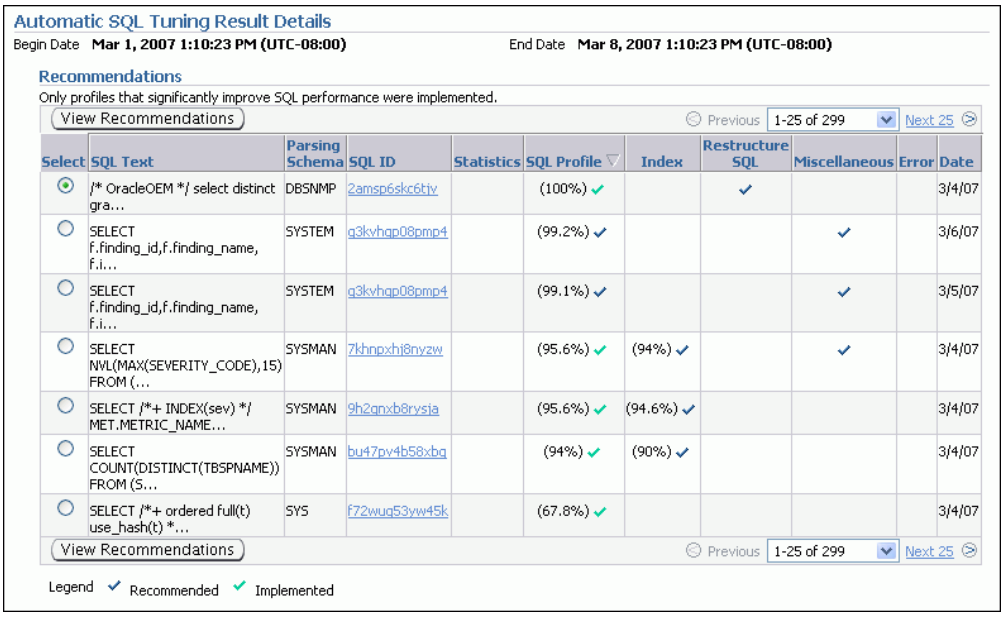
4. Click **Automatic SQL Tuning**.

   The Automatic SQL Tuning Result Summary page appears, showing graphical summaries of the Automatic SQL Tuning Advisor activities and findings.



5. To view recommendations, click **View Report** under the heading Task Activity Summary.

   The Automatic SQL Tuning Result Details page appears, showing the SQL statements for which recommendations were made during the designated period.



A green check mark in the SQL Profile column indicates a recommendation that was automatically implemented.

By default, automatic implementation is disabled. See "Configuring the Automatic SQL Tuning Advisor" on page 10-20 for instructions for enabling it.

6. (Optional) Select a SQL statement in the Recommendations table (based on the SQL Text column), and then click **View Recommendations**.

   The Recommendations for SQL ID page appears, describing each recommendation for the statement in detail. On this page, you can select a recommendation and then click **Implement** to implement it.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*
- "About the Automatic SQL Tuning Advisor" on page 10-19

Running the SQL Tuning Advisor

Use the SQL Tuning Advisor for tuning SQL statements. Typically, you run this advisor in response to an ADDM performance finding that recommends its use. You can also start the SQL Tuning Advisor manually. One reason to do this is to tune statements that the Automatic SQL Tuning Advisor has not considered for tuning.

As described in "About the SQL Advisors" on page 10-18, the SQL Tuning Advisor can select SQL statements to tune from a number of sources. The following scenario assumes that you want to tune the SQL statements with the most activity:

To run the SQL Tuning Advisor:

1. Go to the Database Home page.

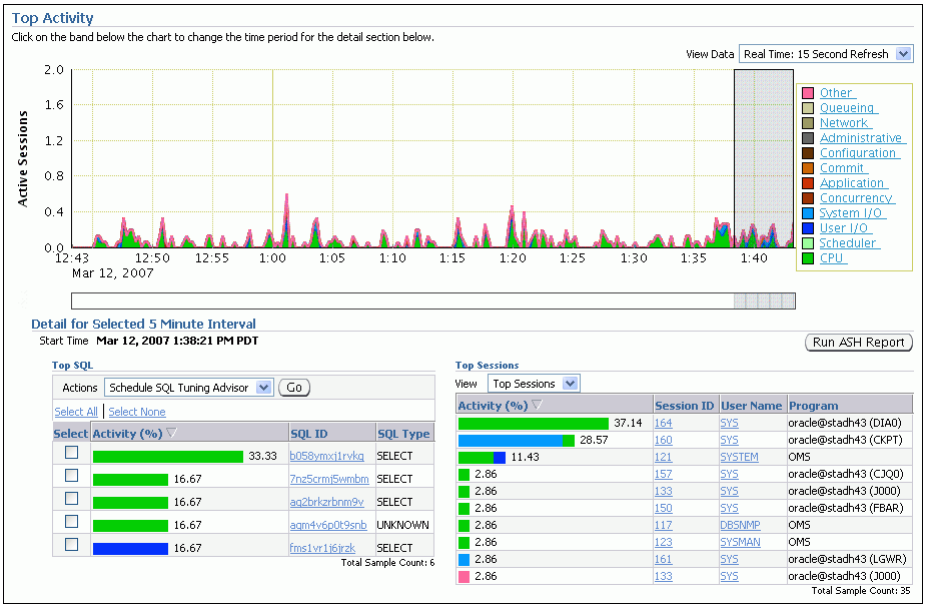
   See "Accessing the Database Home Page" on page 3-4.

2. At the bottom of the page, under Related Links, click **Advisor Central**.
3. On the Advisor Central page, click **SQL Advisors**.
4. On the SQL Advisors page, click **SQL Tuning Advisor**.

The Schedule SQL Tuning Advisor page appears.

5. Under SQL Tuning Advisor Data Source Links, click **Top Activity**.

   The Top Activity page appears. This page has a timeline graph showing the database activity based on the active sessions over the last hour.



6. (Optional) Expand the timeline by selecting **Historical** from the View Data list at the upper right-hand side of the page.
7. Select an interval to analyze by clicking the bar under the timeline graph.

   The Top SQL and Top Sessions tables show the activity within the selected period.
8. In the Top SQL section, select one or more SQL statements, and then click the **Go** button next to Schedule SQL Tuning Advisor.

   The Schedule SQL Tuning Advisor page returns, indicating the statements that are to be tuned.
9. Enter a task name and description, select the scope for the analysis (**Comprehensive** or **Limited**), and select a start time for the analysis task.
10. Click **OK**.

    A Processing page appears. When the task is complete, the SQL Tuning Results page appears, showing a summary of recommendations for each SQL statement analyzed.

    The recommendation can include one or more of the following:

    - Accept the generated SQL profile.
    - Gather optimizer statistics on objects with stale or no statistics.
    - Rewrite a query for better performance.
    - Create an index to offer alternate, faster access paths to the query optimizer.
11. To view recommendations for a specified SQL statement, select a statement from the list of recommendations, and then click **View**.

    The Recommendations for SQL ID page appears, showing one or more recommendations for the statement.

12. (Optional) Select a recommendation, and then click **Implement**.

    The Implement Recommendation page appears. Depending on the type of recommendation, select options, and then click **OK** to proceed.

    See Also:

    - "About the SQL Advisors" on page 10-18
    - "About the Automatic SQL Tuning Advisor" on page 10-19
    - *Oracle Database 2 Day + Performance Tuning Guide*

Running the SQL Access Advisor

You run the SQL Access Advisor to get recommendations for improving the performance of a workload. You can run it on a periodic basis to avoid performance problems, or run it to verify schema change recommendations from the SQL Tuning Advisor.

To run the SQL Access Advisor:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

2. At the bottom of the page, under Related Links, click **Advisor Central**.
3. On the Advisor Central page, click **SQL Advisors**.
4. On the SQL Advisors page, click **SQL Access Advisor**.

   A page appears prompting you for initial options. Click **Help** for information about the options for this page.

5. Select an option, and then click **Continue**.
6. The Workload Source page appears. It is the first page of the SQL Access Advisor wizard.

   Click **Help** to obtain help for this and subsequent wizard pages. Follow directions and make the required selections for each wizard page, clicking **Next** when you are ready to proceed to the next page.

7. Continue through the wizard until you reach the Review page. Review your selections and then click **Submit** to start the analysis.

   The Advisor Central page appears, displaying a confirmation message indicating that your task has been started successfully.

8. Click the **Refresh** button (not your browser's Refresh icon) to view the status of your task.
9. When your SQL Access Advisor task has completed, select **View Result**.

   The Result for Task page appears.

   The Summary subpage shows you the potential for improvement under the headings Workload I/O Cost and Query Execution Time Improvement.

   The Recommendations subpage shows the recommendations, if any, for improving performance. A recommendation might consist of, for example, a SQL script with one or more `CREATE INDEX` statements, which you can run by clicking **Schedule Implementation**.

See Also:

- "About the SQL Advisors" on page 10-18
- *Oracle Database 2 Day + Performance Tuning Guide*

Using the Memory Advisors

Adequate physical memory has a significant impact on the performance of your Oracle Database. With its automatic memory management capabilities, Oracle Database can automatically adjust the memory distribution among the various SGA and PGA components for optimal performance. These adjustments are made within the boundaries of the total amount of memory that you allocate to the database.

ADDM periodically evaluates the performance of your database to determine performance problems. If ADDM finds that the current amount of available memory is inadequate and adversely affecting performance, it can recommend that you increase memory allocations. You can select new memory allocations using the Memory Advisors.

Additionally, you can use the Memory Advisors to perform what-if analysis on the following:

- The database performance benefits of adding physical memory to your database
- The database performance impact of reducing the physical memory available to your database

See Also:

- "Managing Memory" on page 5-12
- "About Advisors" on page 10-16

Obtaining Memory Sizing Advice

With the Memory Advisors, you can obtain memory sizing advice as follows:

- If automatic memory management is enabled, you can get advice for setting the target amount of memory to allocate to the Oracle instance.
- If automatic memory management is disabled and automatic shared memory management is enabled, you can gain advice on configuring the target sizes of the SGA and instance PGA.
- If only manual shared memory management is enabled, you can get advice on sizing the shared pool, buffer cache, and instance PGA.

The following steps describe how to obtain memory sizing advice when Automatic Shared Memory Management is enabled.

To obtain memory sizing advice:

1. Go to the Database Home page.

   See "Accessing the Database Home Page" on page 3-4.

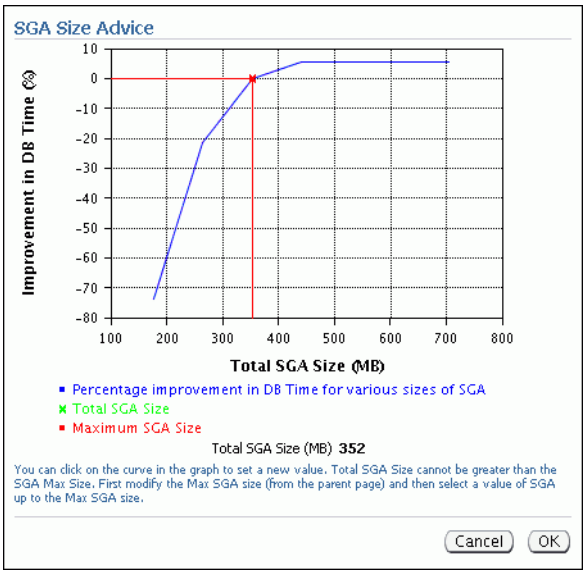
2. At the bottom of the page, click **Advisor Central**.
3. On the Advisor Central page, click **Memory Advisors**.

   The Memory Advisors page appears.

4. On the SGA subpage, next to the Total SGA Size field, click **Advice**.

   The SGA Size Advice child page appears in a separate window. (Figure 10–3).

Figure 10–3 SGA Size Advice



Improvement in DB Time (%) is plotted against Total SGA size. A higher number for Improvement in DB Time is better for performance.

In Figure 10–3, the graph indicates that increasing the SGA size greater than 450 MB results in no performance gain. Thus, 450 MB is the recommended optimal SGA size.

Click **OK** to close the SGA Size Advice child page.

5. Near the top of the Memory Advisors page, click **PGA** to display the PGA subpage.
6. Next to Aggregate PGA Target, click **Advice**.

   The PGA Target Advice page appears, plotting cache hit percentage against PGA target size.

   The cache hit percentage is the percentage of read requests serviced by memory, as opposed to those requests serviced by reading from disk, which is slower. A higher hit percentage indicates better cache performance. The optimal zone for cache hit percentage is between 75 and 100 percent. However, it is not safe to conclude that your database is having performance problems simply because your cache hit percentage is not within the optimal zone. When the amount of currently available PGA memory is not adequate for optimal performance, ADDM automatically recommends adjusting this value in a performance finding.

   Click **OK** to close the PGA Target Advice page.

See Also:

- "Managing Memory" on page 5-12

Monitoring and Tuning: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this section, and includes annotated screenshots.

To view the Monitoring OBE, in your browser, enter the following URL:

```
http://www.oracle.com/technology/obe/11gr1_2day_dba/monitoring/monitoring.htm
```

11

Investigating, Reporting, and Resolving Problems

Oracle Enterprise Manager Database Control (Database Control) includes a feature called the Support Workbench that enables you to view and investigate critical database errors, report these errors to Oracle Support Services, and in some cases, resolve the errors.

This chapter explains critical errors and describes how to use the Database Home page and the Support Workbench to do the following:

- View critical error alerts
- View diagnostic data for critical errors
- Package diagnostic data for upload to Oracle Support Services
- Create and track a service request
- Repair some classes of critical errors

This chapter contains the following sections:

- About Database Problems and Incidents
- Investigating, Reporting, and Resolving a Problem
- Viewing Problems Using the Enterprise Manager Support Workbench

About Database Problems and Incidents

To help diagnosis and resolution of critical errors, Oracle Database Release 11*g* introduces two concepts for Oracle Database: problems and incidents.

A **problem** is a critical error in the database. Critical errors manifest as internal errors, such as `ORA-00600`, and other severe errors, such as `ORA-07445` (operating system exception) or `ORA-04031` (out of memory in the shared pool portion). Problems are tracked in the **Automatic Diagnostic Repository** (**ADR**). The ADR is a file-based repository for storing diagnostic data. Because this repository is stored outside the database, the diagnostic data is available even when the database is down. As of Release 11*g*, the alert log, all trace and dump files, and other diagnostic data are also stored in the ADR.

Each problem has a **problem key**, which is a text string that describes the problem. The problem key includes the error code (such as `ORA 600`), and in some cases, one or more error parameter values or other information.

An **incident** is a single occurrence of a problem. When a problem occurs multiple times, an incident is created for each occurrence. Incidents are timestamped and tracked in the Automatic Diagnostic Repository (ADR).

Each incident is identified by a numeric incident ID, which is unique within the ADR. When an incident occurs, the database performs the following steps:

- Makes an entry in the alert log.
- Sends an incident alert to Enterprise Manager.
- Gathers first-failure diagnostic data about the incident (such as trace files).
- Tags the diagnostic data with the incident ID.
- Stores the data in an ADR subdirectory created for that incident.

Each incident has a problem key and is mapped to a single problem.

Diagnosis and resolution of a critical error usually starts with an incident alert. The incident alert is displayed on the Enterprise Manager Database Home page. You can then view the problem and its associated incidents with Enterprise Manager.

See Also:

- *Oracle Database Administrator's Guide* for more information about the ADR
- "Task 1 – View Critical Error Alerts in Enterprise Manager" on page 11-4

Investigating, Reporting, and Resolving a Problem

This section describes the typical set of tasks that you perform to investigate and report a problem (critical error), and in some cases, resolve the problem. The section begins with a *roadmap* that summarizes these tasks.

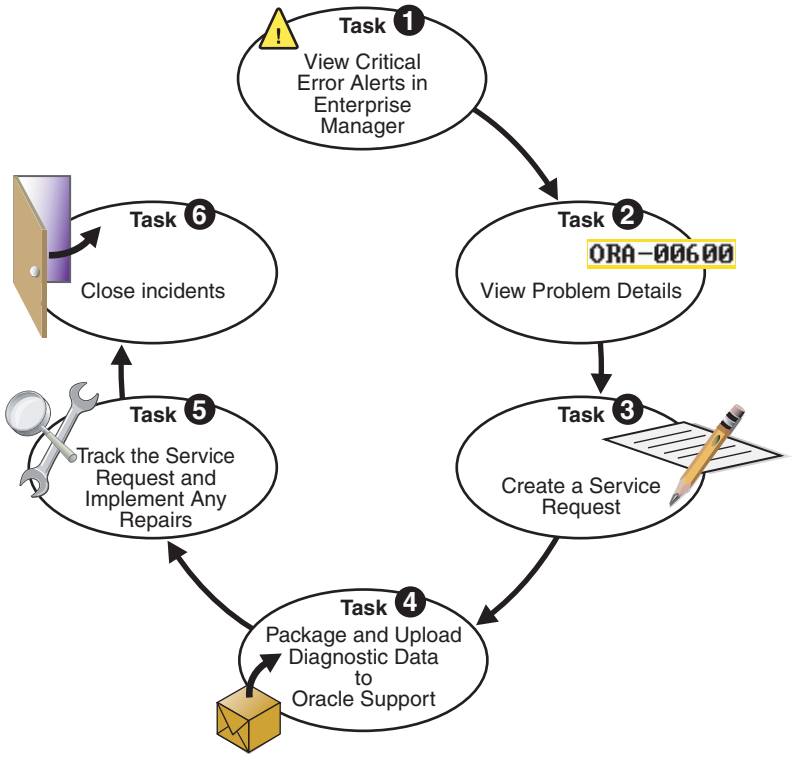
Note: The workflow described in this section includes only the minimum tasks that are required to investigate, report, and in some cases, repair a problem. See *Oracle Database Administrator's Guide* for a more complete workflow that includes additional diagnostics-gathering activities and data customization activities that you can do before uploading the diagnostic data to Oracle Support Services. In some cases, these additional activities may result in a shorter time to problem resolution.

Roadmap: Investigating, Reporting, and Resolving a Problem

You can begin investigating a problem by starting from the Support Workbench Home page. However, the more typical workflow begins with a critical error alert on the Database Home page. This documentation provides an overview of that workflow.

Figure 11–1 illustrates the basic tasks that you complete when encountering a problem.

Figure 11–1 Workflow for Investigating, Reporting, and Resolving a Problem



The following are task descriptions. Subsequent sections provide details for each task.

- **Task 1 – View Critical Error Alerts in Enterprise Manager** on page 11-4

  Start by accessing the Database Home page in Enterprise Manager, and reviewing critical error alerts. Select an alert for which to view details. From the alert details page, go to the Problem Details page.

- **Task 2 – View Problem Details** on page 11-5

  Examine the problem details and view a list of incidents that were recorded for the problem. Display findings from any health checks that were automatically run.

- **Task 3 – (Optional) Create a Service Request** on page 11-6

  Create a service request using Oracle*MetaLink* and optionally record the service request number with the problem information. If you skip this step, you can create a service request later, or the Support Workbench can create one for you.

- **Task 4 – Package and Upload Diagnostic Data to Oracle Support Services** on page 11-7

  Invoke a guided workflow (a *wizard*) that automatically packages all gathered diagnostic data for a problem and uploads the data to Oracle Support Services.

- **Task 5 – Track the Service Request and Implement Any Repairs** on page 11-8

  Optionally maintain an activity log for the service request in the Support Workbench. If appropriate, run Oracle advisors to help repair SQL failures or corrupted data.

- **Task 6 – Closing Incidents** on page 11-9

  Set the status for one, some, or all incidents for the problem to Closed.

See Also:

- "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9
- "About Database Problems and Incidents" on page 11-1

Task 1 – View Critical Error Alerts in Enterprise Manager

You begin the process of investigating problems (critical errors) by reviewing critical error alerts on the Database Home page.

To view critical error alerts:

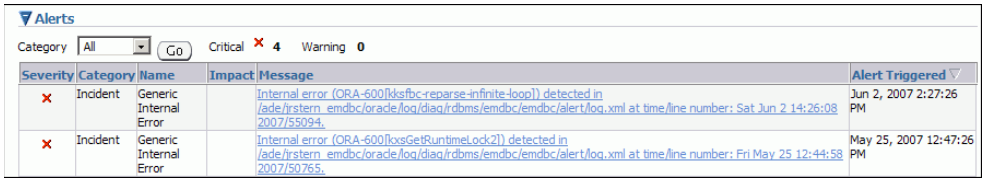
1. Go to the Database Home page in Enterprise Manager.

   See "Accessing the Database Home Page" on page 3-4.

2. In the Alerts section, examine the table of alerts.

   Critical error alerts are indicated by an X in the Severity column, and the text "Incident" in the Category column.

   Note: You may have to click the hide/show arrowhead icon next to the Alerts heading to display the alerts table.

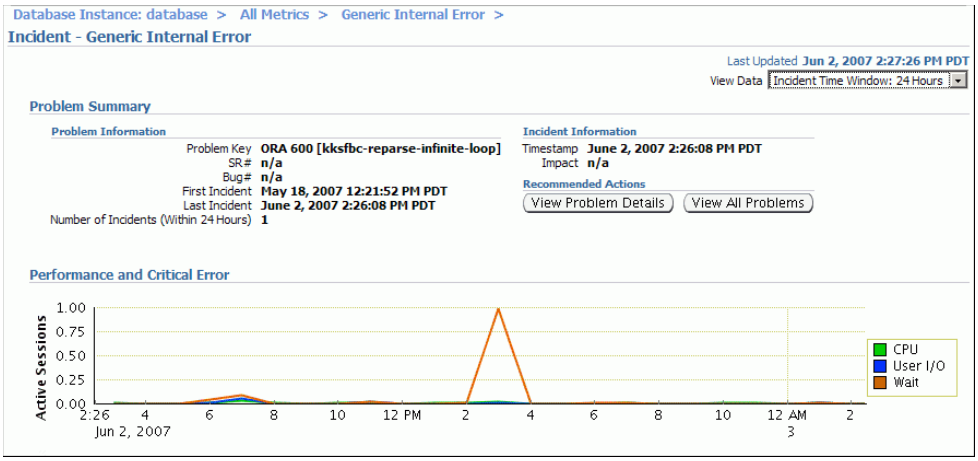


3. (Optional) In the Category list, select **Incident** to view alerts of type Incident only.
4. In the **Message** column, click the message of the critical error alert that you want investigate.

   An Incident detail or Data Failure page appears for the type of incident you selected. For example, if you clicked a message about an ORA-600 error, the Incident - Generic Internal Error page appears.

   This page displays:

   - Problem information, including the number of incidents for the problem
   - A Performance and Critical Error graphical timeline for the 24-hour time period in which the critical error occurred
   - Alert details, including severity, timestamp, and message
   - Controls that enable you to clear the alert or record a comment about it



5. Review the Performance and Critical Error graphical timeline, and note any time correlation between performance issues and the critical error. Optionally clear the alert or leave a comment about it.
6. Perform one of the following actions:
   - If you want to view the details of the problem associated with the critical error alert that you are investigating, proceed with Task 2 – View Problem Details on page 11-5.
   - If the graphical timeline shows a large number of different problems during the 24-hour time period and you want to view a summary of all those problems, complete these steps:
     – Click **View All Problems**.

       The Support Workbench Home page appears.
     – View problems and incidents as described in "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9.
     – Select a single problem and view problem details, as described in "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9.
     – Continue with Task 3 – (Optional) Create a Service Request on page 11-6.

Task 2 – View Problem Details

You continue your investigation with the Problem Details page.

To view problem details:

1. On the Incident detail or Data Failure page, click **View Problem Details**.

   The Problem Details page appears, showing the Incidents subpage.
2. (Optional) To view incident details, in the Incidents subpage, select an incident, and then click **View**.

   The Incident Details page appears, showing the Dump Files subpage.
3. (Optional) On the Incident Details page, click **Checker Findings** to view the Checker Findings subpage.

This page displays findings from any health checks that were automatically run when the critical error was detected.

See Also:

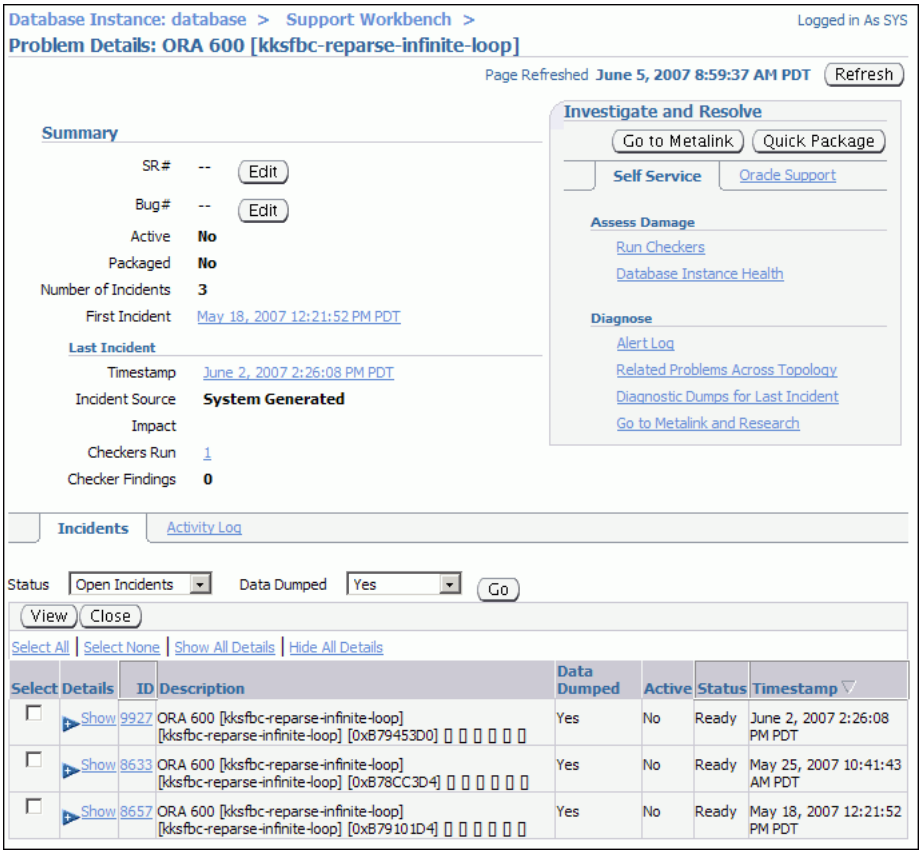
- *Oracle Database Administrator's Guide* for information about health checks and checker findings

Task 3 – (Optional) Create a Service Request

At this point, you can create an Oracle Support service request and record the service request number with the problem information. If you choose to skip this task, the Support Workbench will automatically create a draft service request for you in Task 4.

To create a service request:

1. On the Problem Details page, in the Investigate and Resolve section, click **Go to Metalink**.



   The Oracle*MetaLink* Login and Registration page appears in a new browser window.

2. Log in to Oracle*MetaLink* and create a service request in the usual manner.

   (Optional) Remember the service request number (SR#) for the next step.

3. (Optional) Return to the Problem Details page, and then do the following:

   a. In the Summary section, click the **Edit** button that is adjacent to the SR# label.

   b. In the page that opens, enter the SR#, and then click **OK**.

The SR# is recorded in the Problem Details page. This is for your reference only.

Task 4 – Package and Upload Diagnostic Data to Oracle Support Services

For this task, you use the quick packaging process of the Support Workbench to package and upload the diagnostic information for the problem to Oracle Support Services. Quick packaging has a minimum of steps, organized in a guided workflow (a wizard). The wizard assists you with creating an incident package (referred to as a *package*) for a single problem, creating a ZIP file from the package, and uploading the file. With quick packaging, you are not able to edit or otherwise customize the diagnostic information that is uploaded. Using quick packaging is the direct, straightforward method to package and upload diagnostic data.

If you want to edit or remove sensitive data from the diagnostic information, enclose additional user files (such as application configuration files or scripts), or perform other customizations before uploading, you must use the custom packaging process. See *Oracle Database Administrator's Guide* for instructions. When you complete those instructions, you may continue with Task 5 – Track the Service Request and Implement Any Repairs on page 11-8.

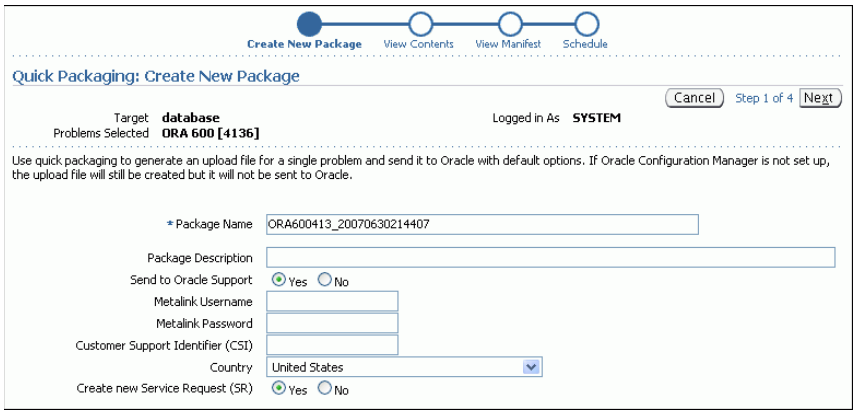
Note: The Support Workbench uses Oracle Configuration Manager to upload the diagnostic data. If Oracle Configuration Manager is not installed or properly configured, the upload may fail. In this case, a message is displayed with a request that you upload the file to Oracle Support manually. You can upload files manually with Oracle*MetaLink*.

For more information about Oracle Configuration Manager, see *Oracle Configuration Manager Installation and Administration Guide*.

To package and upload diagnostic data to Oracle Support:

1. On the Problem Details page, in the Investigate and Resolve section, click **Quick Package**.

   The Create New Package page of the Quick Packaging wizard appears.



2. (Optional) Enter a package name and description.
3. Fill in the remaining fields on the page. If you have already created a service request for this problem, select **No** next to Create new Service Request (SR).

If you select Yes, the Quick Packaging wizard creates a draft service request on your behalf. You must later log in to Oracle*MetaLink* and fill in the details of the service request.

4. Click **Next**, and then proceed with the remaining pages of the Quick Packaging wizard.

   When the Quick Packaging wizard is complete, the package that it creates remains available in the Support Workbench. You can then modify it with custom packaging operations (such as adding new incidents) and reupload the package at a later time.

Task 5 – Track the Service Request and Implement Any Repairs

After uploading diagnostic information to Oracle Support Services, you might perform various activities to track the service request, to collect additional diagnostic information, and implement repairs. Among these activities are the following:

- Adding an Oracle bug number to the problem information.

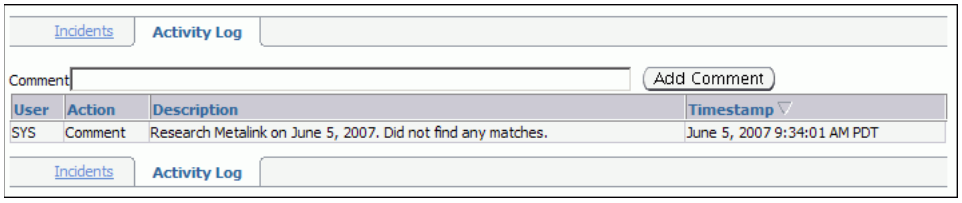
  To do so, on the Problem Details page, click the **Edit** button that is adjacent to the Bug# label. This is for your reference only.

- Adding comments to the problem activity log.

  You may want to do this to share problem status or history information with other DBAs in your organization. For example, you could record the results of your conversations with Oracle Support.

  To add comments to the problem activity log:

  1. Go to the Problem Details page for the problem, as described in "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9.
  2. Click **Activity Log** to display the Activity Log subpage.



  3. In the **Comment** field, enter a comment, and then click **Add Comment**.

     Your comment is recorded in the activity log.

- If a new incident occurs, adding them to the package and reuploading to Oracle Support Services.

  For this activity, you must use the custom packaging method described in *Oracle Database Administrator's Guide*.

- Running health checks.

  See *Oracle Database Administrator's Guide* for information about health checks.

- Running a suggested Oracle advisor to implement repairs.

  You can access the suggested advisor in one of the following ways:

  – **Problem Details page**—In the Self-Service tab of the Investigate and Resolve section

- **Support Workbench Home page**—On the Checker Findings subpage
- **Incident Details page**—On the Checker Findings subpage

Table 11–1 lists the advisors that help repair critical errors.

Table 11–1 Oracle Advisors that Help Repair Critical Errors

Advisor	Critical Errors Addressed	See
Data Recovery Advisor	Corrupted blocks, corrupted or missing files, and other data failures	"Performing Oracle Advised Recovery" on page 9-22
SQL Repair Advisor	SQL statement failures	*Oracle Database Administrator's Guide*

See Also:

- "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9

Task 6 – Closing Incidents

When a particular incident is no longer of interest, you can close it. By default, closed incidents are not displayed on the Problem Details page.

All incidents, whether closed or not, are purged after 30 days. You can disable purging for an incident on the Incident Details page.

To close incidents:

1. Go to the Support Workbench Home page.

   See "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-9.

2. Select the desired problem, and then click **View**.

   The Problem Details page appears.

3. Select the incidents to close and then click **Close**.

   A confirmation page appears.

4. Enter an optional comment and then click **OK**.

Viewing Problems Using the Enterprise Manager Support Workbench

You use the Enterprise Manager Support Workbench home page to view all problems or only those within a specified time period.

To access the Support Workbench home page:

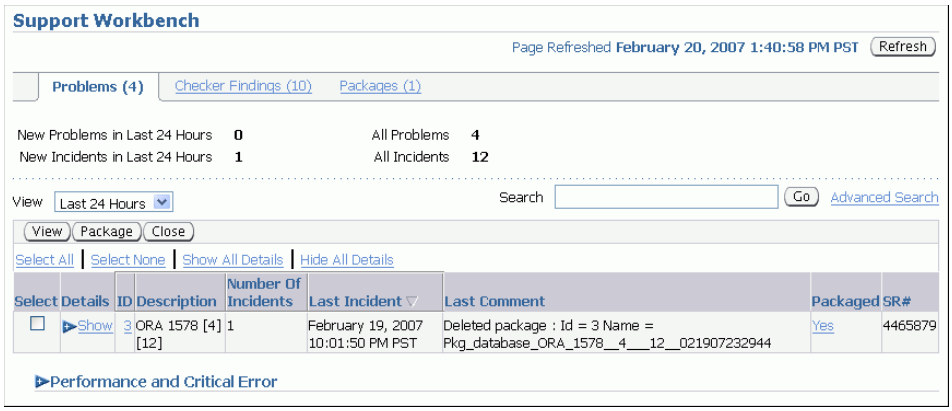
1. Go to the Database Home page in Enterprise Manager.

   See "Accessing the Database Home Page" on page 3-4.

2. Click **Software and Support** to view the Software and Support page.

3. In the Support section, click **Support Workbench**.

   The Support Workbench home page appears, showing the Problems subpage. By default the problems from the last 24 hours are displayed.



4. To view all problems, select **All** from the View list.
5. (Optional) If the Performance and Critical Error section is hidden, click the **Show/Hide** icon adjacent to the section heading to show the section.

   This section enables you to view any correlation between performance changes for your database and incident occurrences.
6. (Optional) Under the Details column, click **Show** to display a list of all incidents for a problem, and then click an incident ID to display the Incident Details page.

To view details for a particular problem:

1. On the Support Workbench home page, select the problem, and then click **View**.

   The Problem Details page appears, showing the Incidents subpage.
2. (Optional) To view closed incidents as well as open incidents, select **All** from the View list.
3. (Optional) To view details for an incident, select the incident, and then click **View**.

   The Incident Details page appears.
4. (Optional) To view checker findings for the incident, on the Incident Details page, click **Checker Findings**.

   The Checker Findings subpage appears.
5. (Optional) On the Incident Details page, to view the user actions that are available to you for the incident, click **Additional Diagnostics**. Each user action provides a way for you to gather additional diagnostics for the incident or its problem.

12

Managing Oracle Database Software

This chapter describes how to keep your Oracle Database software up-to-date with patch and software releases. This chapter contains the following sections:

- About Software Management and Patch Releases
- Patching the Oracle Software
- Viewing Critical Patch Advisories
- Upgrading a Database
- Managing Oracle Software: Oracle By Example Series

About Software Management and Patch Releases

Software management involves keeping your Oracle Database software up-to-date with the latest product fixes. When a product defect, or a **bug**, is discovered, a patch is created to fix the problem. A **patch** corrects a single defect in the installed software. Individual patches, also referred to as **interim patches**, are made available to customers who for business reasons cannot wait until the next patch set to receive the product fix.

Oracle periodically issues maintenance releases for its software, in the form of patch sets. A **patch set** is a collection of product fixes that have been released up to the time of the maintenance release. Patch sets are fully tested and integrated product fixes. All the product fixes in the patch set have been tested and are certified to work with each other.

Every patch or patch set is associated with a bug number for identification purposes. Patch sets are also associated with version numbers. For example, if you use Oracle Database 11*g* Release 11.1.0.1, then an available patch set might be 11.1.0.3. The version number of the patched software does not change if an interim patch is applied.

Every patch has an associated README file that describes how it fixes the software. The README file also has instructions for applying the patch.

Every patch set is accompanied by a Patch Set Notes file that contains installation instructions and information about the product fixes contained within the patch set. When you apply a patch set to your Oracle software, you change the maintenance release number for your installed software. Applying a patch set affects the software residing in your Oracle home only, with no change to the data in the database.

You can use Oracle Enterprise Manager Database Control to automate the staging and application of Oracle patches and patch sets. Database Control stages an Oracle patch by downloading it from the Oracle*MetaLink* Web site and copying it to a directory on the server.

Patching the Oracle Software

The steps involved in patching the Oracle software are as follows:

- Determining Your Oracle Database Software Environment
- Setting Your OracleMetaLink Credentials
- Staging and Applying Patches and Patch Sets

Determining Your Oracle Database Software Environment

To select the appropriate patch set release for your environment, you need to know the following details about your Oracle environment:

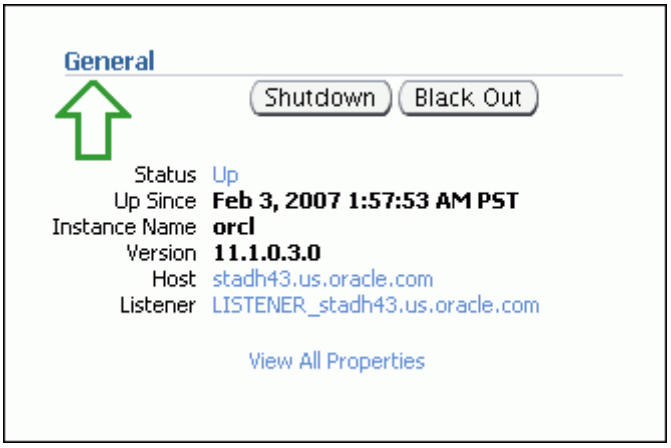
- Oracle Database version
- Oracle home location
- Hardware configuration

Determining Your Oracle Database Version

You can determine your Oracle Database version at the Enterprise Manager Database Control (Database Control) Database Home page.

To determine the Oracle Database version:

- View the Version number under General on the Database Home page.



Determining the Location of Your Oracle Home

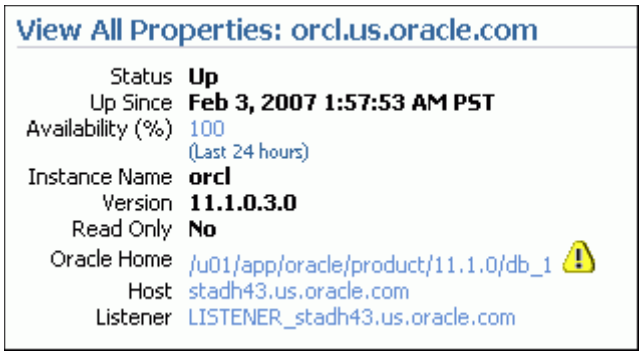
Your Oracle home is the operating system location of your Oracle Database installation.

To determine the location of your Oracle home:

1. In the General section of the Database Home page, click **View All Properties**.

   The View All Properties page appears.

   This page displays the path to your Oracle home.

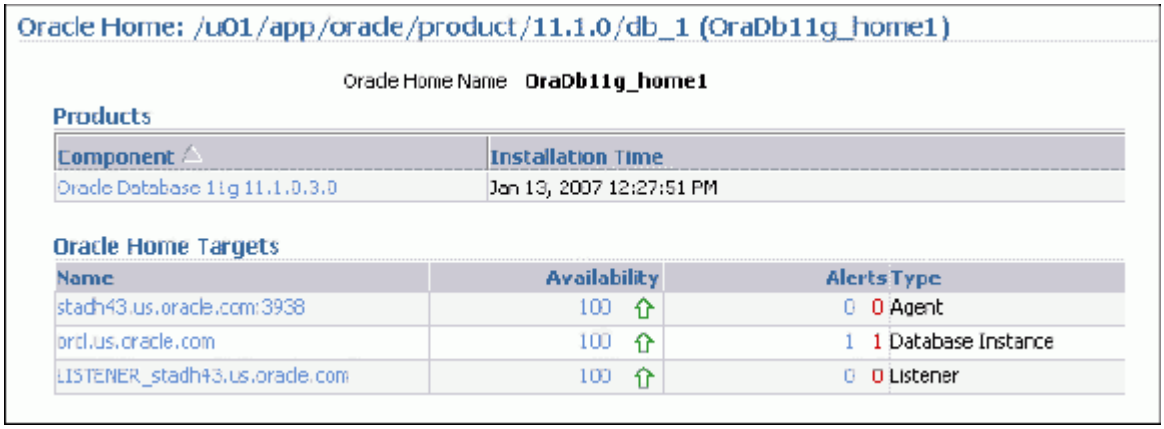


2. Click the path link next to **Oracle Home**.

   The Oracle home: *Oracle_home* page appears, as shown in Figure 12–1.

   This page displays the file directory of your Oracle home, along with the installed Oracle components and time of installation.

Figure 12–1 Oracle Home Page



3. In the Products table, click one of the links in the **Component** column, for example, **Oracle Database 11*g* 11.1.0.3.0**.

   The Oracle Component: *component_name* page appears.

   This page displays the following information about the selected component:

   - Interim patches affecting *component_name* with a description and installation time for each patch
   - Components that *component_name* depends upon and their installation times
   - Components that *component_name* is used by and their installation times

4. In the Oracle Home Targets table, click one of the links in the **Name** column to display pages for the Enterprise Manager agent, database instance, and listener.

Determining Your Hardware Configuration

Your hardware configuration consists of your database computer and operating system. You need this information to select the appropriate Oracle Database patch or patch set.

To view information about your hardware configuration:

1. In the General section of the Database Home page, click the name of your **Host**.

   The Host: *host_name* page appears.

   This page displays the following information about your hardware configuration:

   - Operating system
   - Hardware platform
   - IP address
   - Number of CPUs
   - Memory size (MB)
   - Local file system (GB)

2. In the Configuration section, click the links to drill down to pages that describe your operating system, hardware platform, and local file system in more detail.

Setting Your Oracle*MetaLink* Credentials

You can obtain patches and patch sets from Oracle*MetaLink* at the following URL:

`https://metalink.oracle.com`

To download patches and patch sets from Oracle*Metalink*, you must register using your customer support identifier, which your company obtains when it signs a support contract with Oracle. When you register with Oracle*MetaLink*, you are given a user name and password that enables you to log in to this site from a Web browser.

Before you begin using the patching features of Database Control, you must configure your Oracle*MetaLink* credentials. After you specify your Oracle*MetaLink* credentials, an automated process called the RefreshFromMetalink job can search Oracle*MetaLink* every day for patches that are applicable to your installed software. You also receive notification about critical patch advisories.

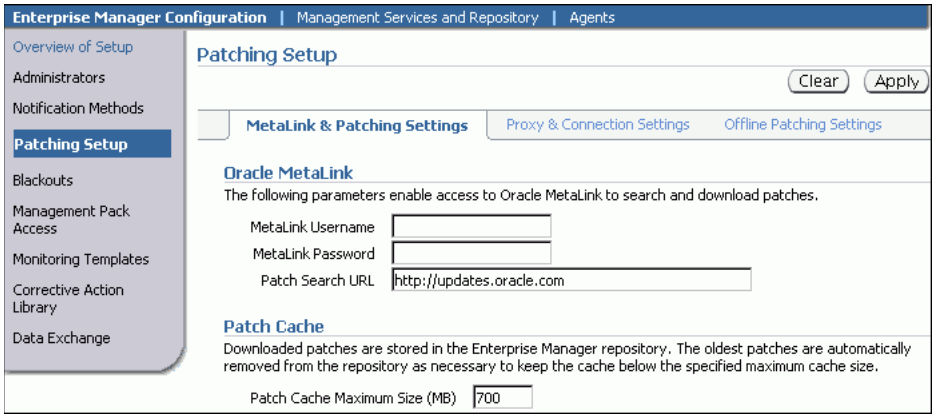
To set your Oracle*MetaLink* login credentials:

1. Click the **Setup** link located at the top and bottom of most Database Control pages.

   The Setup page appears.

2. Click **Patching Setup**.

   The Patching Setup subpage appears.



3. Enter the user name and password for logging in to Oracle*MetaLink* in the **MetaLink Username** and **MetaLink Password** fields.
4. (Optional) Enter the maximum size of your patch cache in the **Patch Cache Maximum Size (MB)** field.

   The patch cache is a temporary area where patches are stored. When the patch cache exceeds the specified maximum size, Database Control automatically performs a purge operation and tries to delete old patches until the patch cache is smaller than the specified maximum size.
5. Click **Apply** to set your Oracle*MetaLink* credentials.

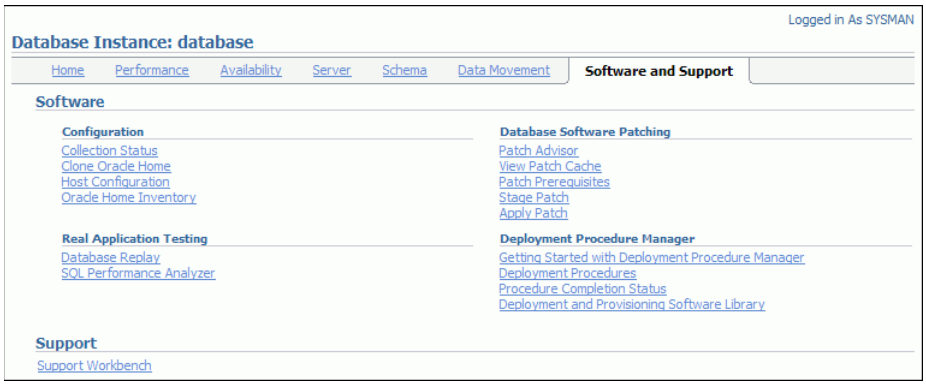
Staging and Applying Patches and Patch Sets

Applying a patch or patch set is the process of installing the *staged*, or locally stored, patch files in the Oracle home on the server. You must specify patching credentials before you can stage and apply a patch or patch set using Enterprise Manager.

To stage and apply patches and patch sets using the Patching wizard:

1. From the Database Home page, click **Software and Support**.

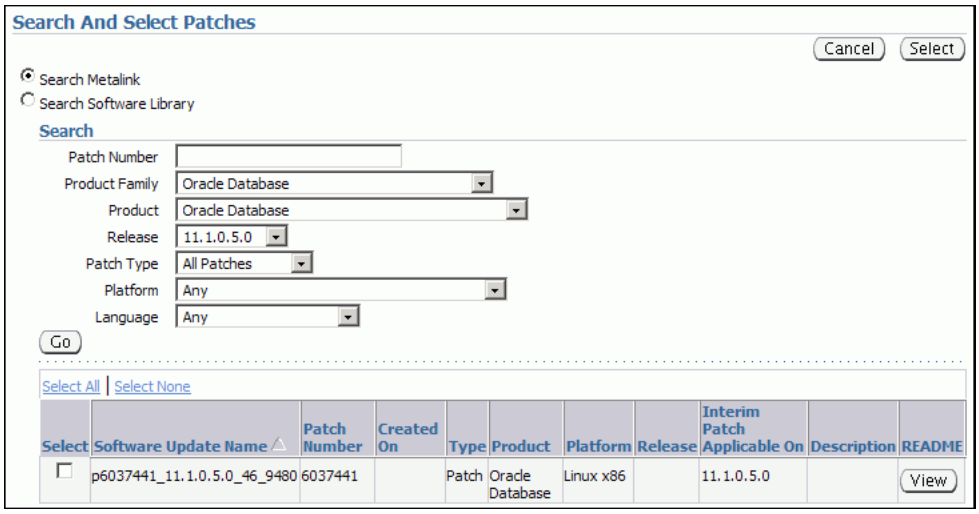
   The Database Instance: *instance_name* page appears.



2. In the Database Software Patching section, select **Apply Patch**.

   The Select Patches page appears.
3. Click **Add Patches** to search for new patches to apply.

   The Search and Select Patches page appears.



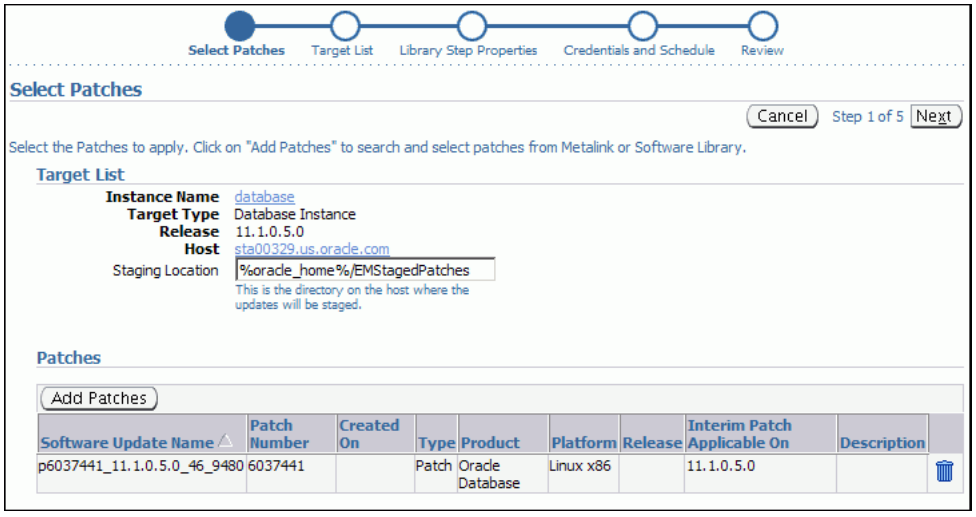
4. In the Search section, choose your current database release from the list of values in the Release list, for example, **11.1.0.5.0**. From the Platform list choose the name that matches your installed operating system, for example, **Linux x86**. After you have made your selections, click **Go**.

   You can optionally limit your search further by choosing different values for **Product Family, Product, Patch Type,** and **Language**, and then clicking **Go**.

5. (Optional) Select a patch or patch set and click **View** to view the patch details.

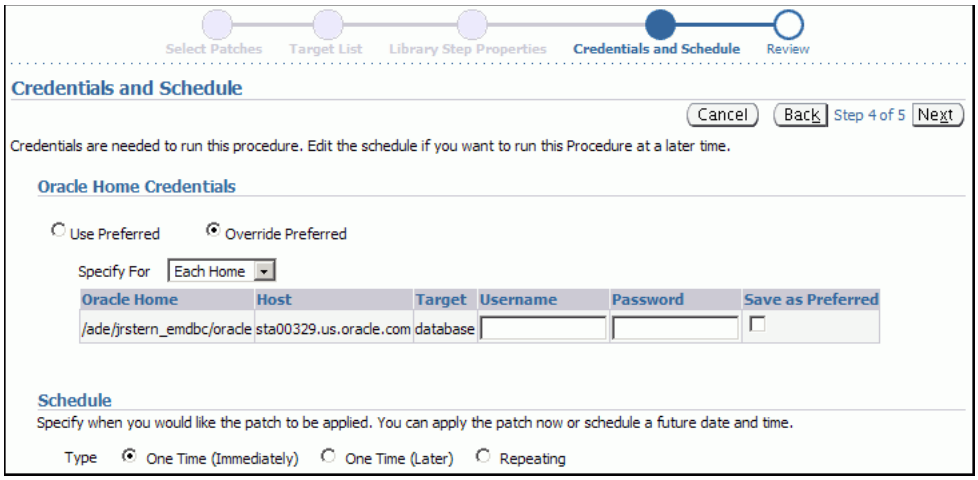
6. Select a patch or patch set and start the Patching wizard by clicking **Select** in the top right corner of the page.

   The Select Patches page appears.



7. If the patch or patch set you selected in Step 5 does not appear in the Patches table, click **Add Patch** and repeat the search to locate the patch. When the patch or patch set you have selected is displayed in the Patches table, click **Next**.

   The Credentials and Schedule page appears.



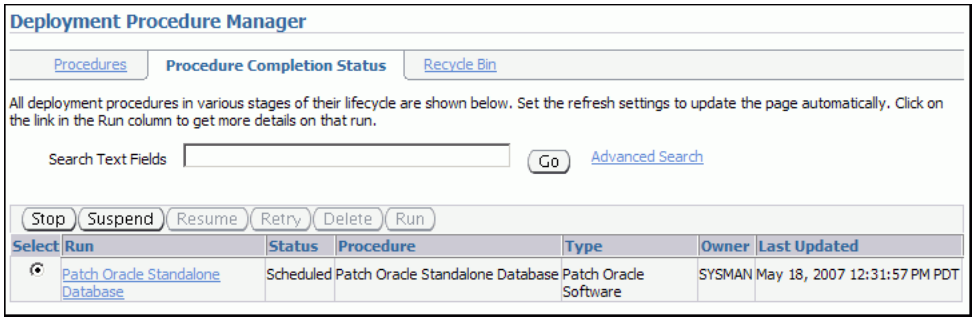
8. Enter the credentials for the operating system user that will perform the patching operation. If you have already configured Preferred Credentials, you can choose the **Use Preferred** option.

   For the **Schedule Type** option, use the default value of **One Time (Immediately)**. Optionally, you can choose to schedule the patching for a later time using the One Time (Later) option. Click **Next** to continue.

   The Review page appears.

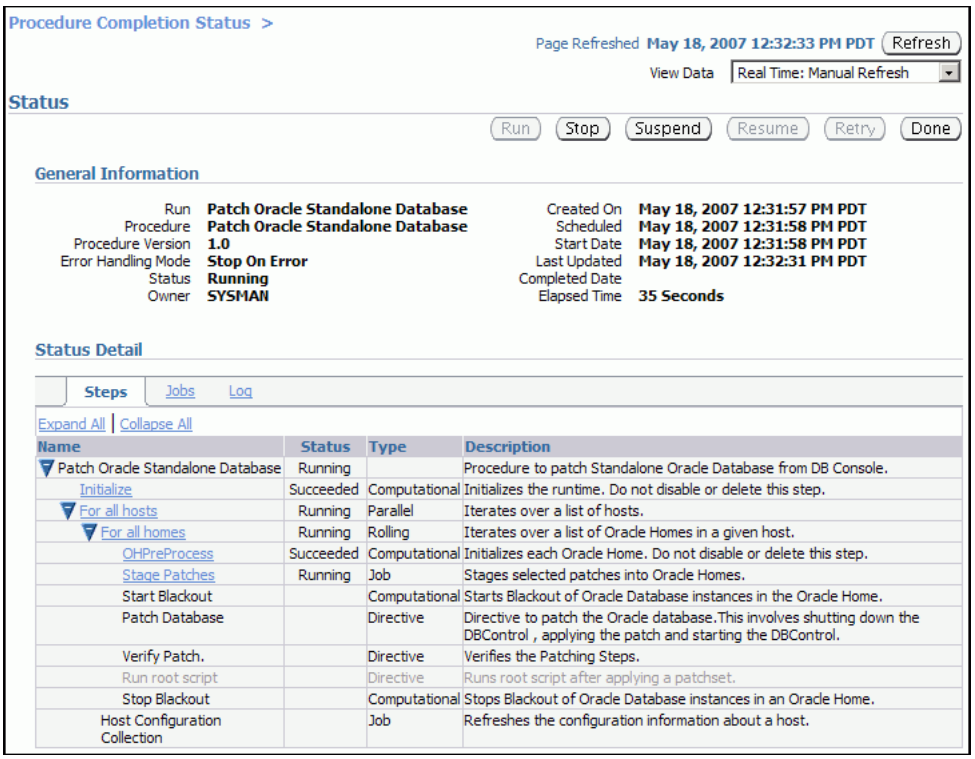
9. Review the information on this page. If any of the information is incorrect, you can click Back to return to a previous page and make corrections. When finished, click **Finish** to submit the job that patches the database.

   The Deployment Procedure Manager page appears.



10. (Optional) To view the progress of the recently submitted patching operation, click **Patch Oracle Standalone Database**.

    The Procedure Completion Status page appears.



As part of the patching procedure, the database instance and Database Control are shut down and then restarted. After they have restarted, you must log in again and navigate to the Procedure Completion Status page to check the status.

11. After you have reviewed the status of each step, you can click **Refresh** to update the display if the procedure has not yet completed.

    When you have finished viewing the results, click **Done**.

12. Click the **Database** tab.

Viewing Critical Patch Advisories

The Patch Advisor in Enterprise Manager describes critical software patches for your installed Oracle products. To help ensure a secure and reliable configuration, all relevant and current critical patches should be applied.

The Patch Advisor provides support for Remedies. When you select an advisory, you can view the calculated remedies from the context of that Advisory, as well as the affected Oracle homes.

The Patch Advisor also displays a list of available patches and patch sets for your installation, along with the name of the feature that is impacted. You can choose to display only patches for features that are used by your database, or all available patches.

To view the critical patch advisories and other recommended patches:

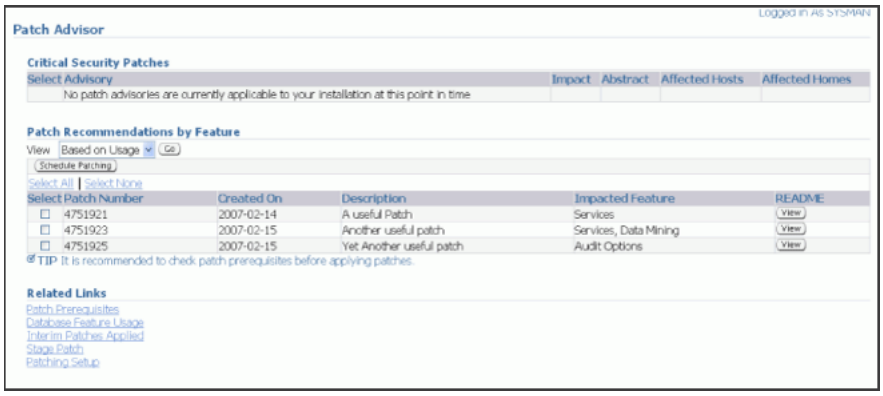
1. Using Database Control, on the Database Home page, in the Policy Violations section, view the count for Critical Security Patches.

   If any critical patches have not been applied to the Oracle home for the database, this section displays the number of critical patch advisories that are relevant to the

Oracle home for the database. Also, a warning icon appears corresponding to the Oracle Home link on the Database Home page is the Oracle home is missing critical patches.

2. To view a list of available critical patch advisories, click the nonzero number next to the heading Critical Security Patches. Alternatively, from the Database Home page, select the **Software and Support** subtab, then, under the heading Database Software Patching, click **Patch Advisor**.

   The Patch Advisor page appears, listing the available critical security patches and patch recommendations by feature.



3. (Optional) In the Critical Security Patches table, click a value in the **Advisory** column to view further details for that critical security patch.
4. (Optional) To view all available patches, in the Patch Recommendations by Feature table, in the View list, select the value **All**, then click **Go**.

Upgrading a Database

Use Database Upgrade Assistant (DBUA) to upgrade an existing database to the current release of Oracle Database.

This section contains these topics:

- Overview of Database Upgrade Assistant
- Database Versions Supported by DBUA
- Starting DBUA
- Upgrading a Database Using DBUA

Overview of Database Upgrade Assistant

Database Upgrade Assistant (DBUA) guides you through the upgrade process and configures your database for the new release. DBUA automates the upgrade process and makes appropriate recommendations for configuration options such as tablespaces and redo logs.

DBUA can be used to upgrade databases created using any edition of the Oracle Database software, including Express Edition (XE) databases.

Pre-Upgrade Checks

DBUA does not begin the upgrade until it completes all of the following pre-upgrade steps:

- Checks for any invalid user accounts or roles
- Checks for any invalid data types or invalid objects
- Checks for any desupported character sets
- Checks for adequate resources, including rollback segments, tablespaces, and free disk space
- Checks for any missing SQL scripts needed for the upgrade
- Backs up all necessary files (optional)

Automated Upgrade Tasks

After completing the pre-upgrade steps, DBUA automatically performs the following tasks:

- Modifies or creates new required tablespaces
- Invokes the appropriate upgrade scripts
- Archives redo logs
- Disables archiving during the upgrade phase (to improve performance)

While the upgrade is running, DBUA shows the upgrade progress for each component. DBUA writes detailed trace and log files and produces a complete HTML report for later reference. To enhance security, DBUA automatically locks new user accounts in the upgraded database. DBUA then proceeds to create new configuration files (initialization parameter and listener files) in the new Oracle home.

Support for Oracle Real Application Clusters

DBUA is fully compliant with Oracle Real Application Clusters (Oracle RAC) environments. In Oracle RAC environments, DBUA upgrades all database and configuration files on all nodes in the cluster.

Support for Automatic Storage Management

DBUA supports upgrades of databases that use Automatic Storage Management (ASM). If an ASM instance is detected, then you have the choice of updating both the database and ASM or only the ASM instance.

Support for Silent Mode

DBUA supports a silent mode of operation in which no user interface is presented to the user. Silent mode enables you to use a single statement for the upgrade.

Database Versions Supported by DBUA

DBUA supports the following versions of Oracle Database for upgrading to Oracle Database 11*g* Release 1 (11.1):

- Oracle9*i* Release 2 (9.2.0.4) and beyond
- Oracle Database 10*g* Release 1 (10.1)
- Oracle Database 10*g* Release 2 (10.2)

If your database version is not in this list, then you need to upgrade first to the closest release listed. You can then upgrade the database to Oracle Database 11*g* Release 1 (11.1).

Starting DBUA

If you install the Oracle Database software only and specify that you are upgrading an existing database to the new Oracle Database release, then DBUA is launched automatically at the end of the software installation. You can then continue as described in "Upgrading a Database Using DBUA" on page 12-12.

If you perform a software-only installation and do not upgrade your database at that time, then you can do so later by launching DBUA.

Be aware of the following before you begin using DBUA:

- You must run Net Configuration Assistant (NETCA) before running DBUA.
- If the database instance is not running, then DBUA will try to start the instance with the default initialization parameter file. If that fails, then you will be prompted to provide the name of the correct initialization parameter file or to start the instance. If the instance is already up and running, then DBUA connects to it.
- If you stop the upgrade, but do not restore the database, then you should not restart DBUA until you start up the existing database in `UPGRADE` mode using the Oracle Database 11*g* software. You cannot go back to the original server unless you restore your database.

  If you restore your database manually (not using DBUA), then remove the following file before starting DBUA:

  ```
  $11.1OracleHome/cfgtoollogs/dbua/logs/Welcome_<SID>.txt
  ```

  The presence of this file indicates to DBUA that this is a rerun operation.

To start DBUA on Microsoft Windows:

- Click **Start**, then select **Programs**, then **Oracle - *HOME_NAME***, then **Configuration and Migration Tools**, and then select **Database Upgrade Assistant**.

  The Database Upgrade Assistant: Welcome window appears.

To start DBUA on any supported platform:

1. Open a command window.
2. (Linux and UNIX systems only) Set the required environment variables by running the script `oraenv` (for the Bourne, Korn, or Bash shells) or `coraenv` (for the C shell).

   These scripts are typically located in the `/usr/local/bin` directory. They may prompt you for the desired value of the `ORACLE_SID` environment variable. If so, supply the system ID (SID) that you chose when you installed Oracle Database. The default SID that the installer typically suggests is `orcl`.
3. (Linux and UNIX systems only) Either ensure that the *`Oracle_home`*`/bin` directory is in your `PATH` environment variable, or change directory to *`Oracle_home`*`/bin`.
4. Enter the following command:

   ```
   dbua
   ```

The Database Upgrade Assistant: Welcome window appears.

Note: The `dbua` executable is typically found in your *`Oracle_home`*`/bin` directory.

See Also: *Oracle Database Upgrade Guide* for more information about Database Upgrade Assistant

Upgrading a Database Using DBUA

Complete the following steps to upgrade a database using DBUA. If you need help at any window or want to consult more documentation about DBUA, then click the **Help** button to access the online Help.

To upgrade a database using DBUA:

1. Start DBUA. See "Starting DBUA" on page 12-11.
2. At the Welcome window of DBUA, make sure the database being upgraded meets the specified conditions. Then, click **Next**.

   If an ASM instance is detected on the system, then the Upgrade Operations window appears. If no ASM instance is detected, then the Databases window appears.
3. (ASM detected only) From the Upgrade Operations window you can choose to upgrade only the ASM instance or the database. If you choose to upgrade the database and if the database is using ASM, then DBUA will ask you whether or not to upgrade the ASM instance along with the database. Oracle recommends that you upgrade the database and ASM in separate DBUA sessions, in separate Oracle homes.
4. At the Databases window, select the database you want to upgrade from the Available Databases table. Then, click **Next**.

   You can select only one database at a time. If you are running DBUA from a user account that does not have `SYSDBA` privileges, then enter the user name and password credentials to enable `SYSDBA` privileges for the selected database.
5. DBUA displays a message saying it is getting database information. DBUA analyzes the selected database, performing pre-upgrade checks and displaying warnings as necessary:

   - It checks for any online redo log files of a size less than 4 megabytes (MB). If such files are found, then DBUA gives the option to drop or create new online redo log files.
   - It checks the initialization parameter file for any obsolete or deprecated initialization parameters.

   If no problems are found, then the Diagnostic Destination window appears.
6. In the **Diagnostic Destination** field, do one of the following:

   - Leave the setting at its default value, which is the Oracle base directory.
   - Enter a new destination.
   - Click **Browse** and select a new destination.

Diagnostic Destination is the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination, user dump destination and core dump destination from earlier Oracle Database releases.

Click **Next**.

The Move Database Files window appears.

7. Do one of the following:
   - Select **Do Not Move Database Files as Part of Upgrade**.
   - Select **Move Database Files During Upgrade**.

     If you choose to move database files, then you must also select either File System or Automatic Storage Management (ASM).

   Click **Next**.

   The Recompile Invalid Objects window appears.

8. (Optional) Select **Recompile invalid objects at the end of upgrade** and modify the value of degree of parallelism.

   When you upgrade your database to the new Oracle Database release, many of the PL/SQL modules in the database become invalid. By default, Oracle Database will recompile invalid PL/SQL modules as they are used, but this takes time and can result in poor performance. To eliminate these performance issues, select **Recompile invalid objects at the end of upgrade**. All the invalid PL/SQL modules will be recompiled immediately after the upgrade is performed. The task of recompiling all the invalid PL/SQL modules in your database can take a significant amount of time and increase the time it takes to complete your database upgrade.

   If you have multiple CPUs, then DBUA automatically adds a **Degree of Parallelism** menu to the Recompile Invalid Objects window. Parallel processing reduces the time it takes to recompile all the invalid PL/SQL modules in your database. DBUA automatically sets the degree of parallelism to one less than the number of available CPUs. You can select a different value from the menu.

   Note: Selecting **Recompile invalid objects at the end of upgrade** is equivalent to running the *Oracle_home*`/rdbms/admin/utlrp.sql` script, which is used to recompile stored PL/SQL and Java code.

   Click **Next**.

   The Backup window appears.

9. Select one of the following options:
   - **I have already backed up my database.**
   - **I would like this tool to back up the database.**

     If you use DBUA to back up your database, then DBUA makes a copy of all database files in the directory that you specify in the **Backup Directory** field. DBUA performs this consistent backup automatically after it shuts down the database and before it begins the upgrade process. The backup does not compress the database files.

In the **Backup Directory** field, do one of the following:

- Leave the setting at its default value
- Enter a different valid file system path. You cannot specify a raw device for the backup files.
- Click **Browse** and select a new backup destination.

Oracle strongly recommends that you back up your database before starting the upgrade. If errors occur during the upgrade, then you may need to restore the database from the backup.

In addition to creating a backup of your database, DBUA creates an executable script in the directory specified in the Backup Directory field. If needed, you can use the executable script appropriate to your system to restore the database files:

- Microsoft Windows: *db_name*`_restore.bat`
- Linux or UNIX systems: *db_name*`_restore.sh`

Click **Next**.

The Recovery Configuration window appears.

10. In the **Flash Recovery Area** field, do one of the following:
    - Leave the setting at its default value
    - Enter a different Flash Recovery Area
    - Click **Browse** and select a different Flash Recovery Area.

    The Flash Recovery Area can be used to recover data that would otherwise be lost during a failure. This location is also used by Enterprise Manager if you have enabled local management and daily backups on the Database Control Management Options page.

    In the **Flash Recovery Area Size** field, select the units you want from the list and do one of the following:
    - Leave the setting at its default value.
    - Enter a different value for Flash Recovery Area Size.
    - Use the up and down arrows to set a different Flash Recovery Area Size.

    In the Recovery Configuration window, you specify a Flash Recovery Area and enable archiving. It is important to configure these features for your database so you can recover your data in the event of a failure.

    Click **Next**.

    The Summary window appears.

11. Review the list the initialization parameters that will be set for the database during the upgrade. Click **Back** to correct any errors, or click **Finish**.

    Note: The database will not be available for general use during the upgrade process that begins when you click **Finish**.

    A Progress window appears and DBUA begins to perform the upgrade.

12. You might encounter error messages with Ignore and Abort choices:

- **Ignore** - Ignores the error, skips the current step, and proceeds with the upgrade. The ignored errors are logged and shown later in the summary. After the upgrade is complete, you can fix the problem, restart DBUA, and complete the skipped steps.
- **Abort** - Stops the upgrade process. DBUA prompts you to restore the database if the database backup was performed by DBUA. After the database has been restored, correct the error and restart DBUA to perform the upgrade again. If you do not restore the database, then DBUA leaves the database in its current state so that you can proceed with a manual upgrade.

After the upgrade has completed, the following message is displayed:

```
Upgrade is complete. Click "OK" to see the results of the upgrade.
```

Click **OK**.

The Upgrade Results window appears.

13. Examine the results of the upgrade. The upgrade results summary describes the original and upgraded databases and changes made to the initialization parameters.

    The upgrade results also include an Upgrade Details section that describes the steps performed during the database upgrade. This section provides each step name, the log file for the step, and the status. In some cases, you can click the status to display details about the execution step. The Upgrade Details section also includes the directory where the various log files are stored after the upgrade. You can examine any of these log files to obtain more details about the upgrade process.

    Note: An HTML version of the Upgrade Results is also saved in the log files directory.

14. Click **Configure Database Passwords**.

    The Password Management dialog box appears.

15. The Password Management dialog box enables you to change the default password for a user after you upgrade the database.

    Note: To prevent unauthorized use of the database, change all user passwords immediately after you upgrade your database.

    Click the **Lock Account?** column for a user to lock or unlock an account. A check mark indicates that the account is locked.

    Enter a new password for a user in its **New Password** column.

    Confirm the new password by entering it in the **Confirm Password** column.

    Click **OK** to return to the Upgrade Results window.

16. Click **Restore** if you are not satisfied with the upgrade results.

    Depending on the method you used to back up your database, the restore operation performs one of the following tasks:

    - If you used DBUA to back up your database, then clicking Restore copies the original database files and the original database settings from the backup.

- If you used your own backup procedure to back up the database, then clicking Restore copies only the original database settings. To restore the database itself, copy the datafiles from the backup you created using your own backup utilities.

Click **Close** to quit DBUA if you are satisfied with the upgrade results.

DBUA removes the entry of the upgraded database from the old `listener.ora` file and restarts the listener.

WARNING: If you retain the old Oracle Database software, then never start the upgraded database with it. Only start the database with the executable files in the new Oracle Database installation. Also, before you remove the old Oracle Database environment, make sure you relocate any datafiles in that environment to the new Oracle Database environment. See *Oracle Database Administrator's Guide* for information about relocating datafiles.

See Also: *Oracle Database Upgrade Guide* for information about additional tasks that should be completed after upgrading a database

Managing Oracle Software: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter, and includes annotated screenshots.

To view the Managing Oracle Software OBE, in your browser, enter the following URL:

`http://www.oracle.com/technology/obe/11gr1_2day_dba/managing/managing.htm`

A

Administering Automatic Storage Management

This appendix discusses using Automatic Storage Management (ASM). ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices. This chapter includes the following topics:

ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices. This section

- About Automatic Storage Management
- Overview of Disks, Disk Groups, and Failure Groups
- Installing Automatic Storage Management
- Accessing the Automatic Storage Management Home Page
- Monitoring Disk Group Space Usage
- Creating a Disk Group
- Dropping a Disk Group
- Adding Disks to a Disk Group
- Dropping Disks from a Disk Group
- Backing Up ASM-Managed Files
- Automatic Storage Management: Oracle By Example Series

See Also:

- *Oracle Database Storage Administrator's Guide* for more information about Automatic Storage Management

About Automatic Storage Management

Automatic Storage Management (ASM) is an integrated, high-performance database file system and disk manager. You use ASM instead of an operating system file system to store your database files.

ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. ASM eliminates the need for you to manage potentially thousands of database files.

ASM groups the disks in your storage system into one or more **disk groups**. You manage a small set of disk groups, and ASM automates the placement of the database files within those disk groups.

ASM provides the following benefits:

- **Striping**—ASM spreads data evenly across all disks in a disk group to optimize performance and utilization. This even distribution of database files eliminates the need for regular monitoring and I/O performance tuning.

  For example, if there are six disks in a disk group, pieces of each ASM file are written to all six disks. These pieces come in 1 MB chunks known as **extents**. When a database file is created, it is **striped** (divided into extents and distributed) across the six disks, and allocated disk space on all six disks increases evenly. When reading the file, file extents are read from all six disks in parallel, greatly increasing performance.

- **Mirroring**—ASM can increase availability by optionally mirroring any file. ASM mirrors at the file level, unlike operating system mirroring, which mirrors at the disk level. Mirroring means keeping redundant copies, or **mirrored copies**, of each extent of the file to help avoid data loss caused by disk failures. The mirrored copy of each file extent is always kept on a different disk from the original copy. If a disk fails, ASM can continue to access affected files by accessing mirrored copies on the surviving disks in the disk group.

  ASM supports **2-way mirroring**, where each file extent gets one mirrored copy, and **3-way mirroring**, where each file extent gets two mirrored copies.

- **Online storage reconfiguration and dynamic rebalancing**—ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk, ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. This redistribution is known as **rebalancing**. It is done in the background and with minimal impact to database performance. When you request to remove a disk, ASM first rebalances the disk group by evenly relocating all file extents from the disk being removed to the other disks in the disk group.

- **Managed file creation and deletion**—ASM further reduces administrative tasks by enabling files stored in ASM disk groups to be **Oracle-managed files**. ASM automatically assigns file names when files are created, and automatically deletes files when they are no longer needed. For information about Oracle-managed files, see *Oracle Database Administrator's Guide*.

Oracle recommends that you use ASM for your database file storage, instead of raw devices or the operating system file system. However, databases can have a mixture of ASM files and non-ASM files. Oracle Enterprise Manager includes a wizard that enables you to migrate non-ASM database files to ASM.

The ASM Instance

ASM is implemented as a special kind of Oracle instance, with its own System Global Area and background processes. The ASM instance is tightly integrated with the database instance. Every server running one or more database instances that use ASM for storage has an ASM instance. In an Oracle Real Application Clusters environment, there is one ASM instance for each node, and the ASM instances communicate with each other on a peer-to-peer basis. Only one ASM instance is required for each node, regardless of the number of database instances on the node.

Administering ASM

You administer ASM with Oracle Enterprise Manager Database Control (Database Control). To administer the ASM instance and ASM disk groups, you must connect to the ASM instance as a user who has been granted the `SYSASM` system privilege. When you create the ASM instance with Oracle Database Configuration Assistant (DBCA), DBCA grants `SYSASM` to user `SYS`.

See Also:

- *Oracle Database Storage Administrator's Guide* for additional information about ASM
- "Accessing the Automatic Storage Management Home Page" on page A-4

Overview of Disks, Disk Groups, and Failure Groups

You configure ASM by creating disk groups that become the default location for files created in the database. The disk group type determines how ASM mirrors files. When you create a disk group, you indicate whether the disk group is a **normal redundancy** disk group (2-way mirroring for most files by default), a **high redundancy** disk group (3-way mirroring), or an **external redundancy** disk group (no mirroring by ASM). You use an external redundancy disk group if your storage system already does mirroring at the hardware level, or if you have no need for redundant data. The default disk group type is normal redundancy.

A disk group consists of a grouping of disks that are managed together as a unit. These disks are referred to as **ASM disks**. An ASM disk can be a disk device, a partition, or a network-attached file.

When an ASM instance starts, it automatically *discovers* all available ASM disks. **Discovery** is the process of finding all disks that were prepared for ASM by your system administrator, examining their disk headers, and determining which disks belong to disk groups and which are available for assignment to disk groups. ASM discovers disks in the paths that are listed in an initialization parameter, or if the parameter is `NULL`, in an operating system–dependent default path.

Failure groups define ASM disks that share a common potential failure mechanism. An example of a failure group is a set of small computer system interface (SCSI) disks sharing the same SCSI controller. Failure groups are used to determine which ASM disks to use for storing redundant copies of data. For example, if 2-way mirroring is specified for a file, ASM automatically stores redundant copies of file extents in separate failure groups. Failure groups apply only to normal and high redundancy disk groups. You define the failure groups in a disk group when you create or alter the disk group.

Note: By default, when you create a disk group, every disk in the disk group belongs to exactly one failure group. For most systems, the default failure groups work well to prevent data loss.

See Also:

- *Oracle Database Storage Administrator's Guide* for additional information about failure groups, and for instructions for configuring failure groups

Installing Automatic Storage Management

This documentation assumes that Automatic Storage Management (ASM) is already installed and configured on your single-instance server or on your Oracle Real Application Clusters (Oracle RAC) nodes. For information about installing and initially configuring ASM, see the following Oracle publications:

- For a single-instance server, the Oracle Clusterware Installation Guide for your operating system
- For Oracle RAC, the Oracle Real Application Clusters Installation Guide for your operating system

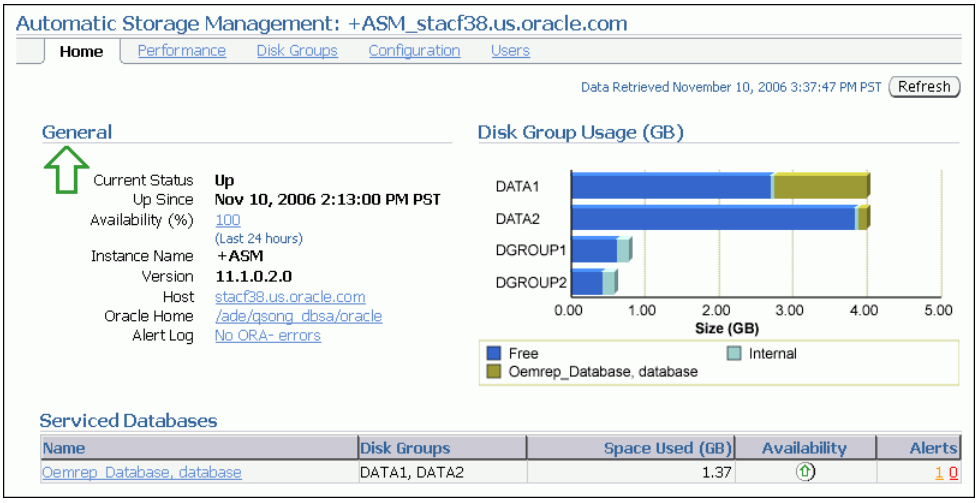
See Also:

- "About Automatic Storage Management" on page A-1

Accessing the Automatic Storage Management Home Page

All Automatic Storage Management (ASM) administration tasks begin with the ASM Home page in Database Control. The ASM Home page displays:

- The status of the ASM instance
- A chart that shows the used and free space of each disk group
- A list of databases that are serviced by the ASM instance



The procedure for accessing the ASM Home page differs depending on whether your database is a single-instance database or an Oracle Real Application Clusters database. Details are provided in the following sections:

- "Accessing the ASM Home Page on a Single-Instance System" on page A-4
- "Accessing the ASM Home Page on an Oracle RAC System" on page A-5

See Also:

- "About Automatic Storage Management" on page A-1

Accessing the ASM Home Page on a Single-Instance System

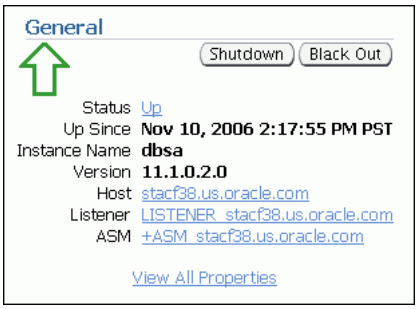
On a single-instance system, you access the ASM Home page starting from the Database Home page.

To access the ASM Home page on a single-instance system:

1. Go to the Database Home page, logging in as any database user.

   See "Accessing the Database Home Page" on page 3-4.

2. Under the General heading, click the link next to the label ASM.



3. If prompted for ASM login credentials, enter the user `SYS`, provide the `SYS` password that was set for the ASM instance upon installation, and connect as `SYSASM`.

Accessing the ASM Home Page on an Oracle RAC System

On an Oracle RAC system, you access the ASM Home page starting from the Cluster Database Home page.

To access the ASM Home page on an Oracle RAC system:

1. Log in to Oracle Enterprise Manager on any node that is running the Oracle Management Service (OMS).

   OMS is automatically started on the node where DBCA was run to create the cluster database. Depending on your configuration, OMS may also be running on other nodes.

   See "Accessing the Database Home Page" on page 3-4.

2. On the Cluster Database Home page, under the Instances heading, click the link for the desired ASM instance.

Monitoring Disk Group Space Usage

Using Oracle Enterprise Manager, you can monitor the total capacity of your disk groups, including the amount of unused space and the amount of unused space that can be safely utilized after taking mirroring into account.

To view Automatic Storage Management (ASM) disk group usage and free space:

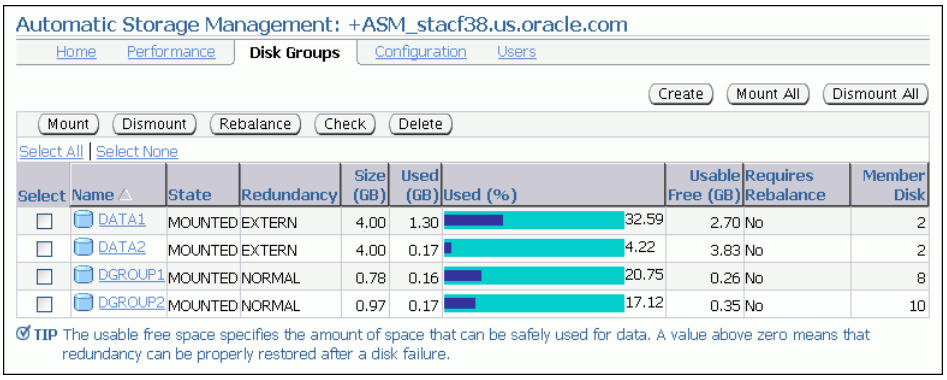
1. Go to the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-4.

2. Click the **Disk Groups** link to view the Disk Groups subpage.

3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the ASM instance was created.

The Disk Groups subpage displays all disk groups with their space usage information.



Note: The Usable Free column displays the space in gigabytes (GB) that is actually available in the disk group. It takes into account the redundancy level of the disk group, and excludes the space that the disk group reserves for restoring full redundancy for all files in the disk group after a disk failure.

See Also:

- "Overview of Disks, Disk Groups, and Failure Groups" on page A-3

Creating a Disk Group

You may want to create additional Automatic Storage Management (ASM) disk groups to do any of the following:

- Have disk groups with different redundancy levels (normal, high, or external), depending on availability requirements and storage system capabilities.
- Separate different classes of storage (for example, SCSI drives and Serial Advanced Technology Attachment (SATA) drives) into different disk groups. Disks in a disk group should have similar size and performance characteristics.
- Store the flash recovery area in a separate disk group from the database.

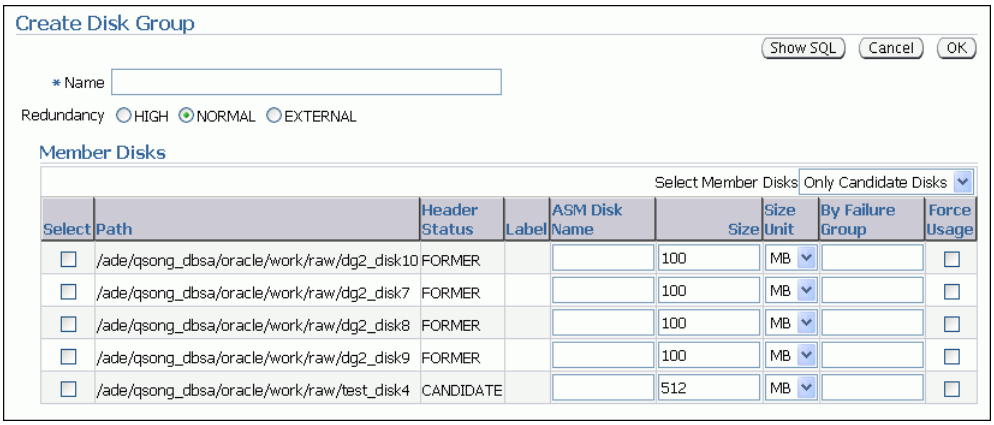
To create a disk group:

1. Go to the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-4.

2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the ASM instance was created.
4. Click **Create**.

   The Create Disk Group page appears. It displays a list of ASM disks that are available to be added to a disk group. This includes disks with the header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`.



5. If you want to display not only disks that are available to be added to a disk group, but all ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), select **All Disks** from the Select Member Disks list.

   The page is refreshed and now displays the complete list of ASM disks that were discovered by the ASM instance.

6. Enter the disk group name, and select redundancy type (high, normal, or external).

7. Select disks as follows:

   a. Select the check box to the left of each disk that you want to include in the new disk group.

   b. If you want to force the inclusion of a disk in the disk group, select the **Force Usage** check box for that disk.

   Note: The Force Usage check box causes the disk to be added to the new disk group even if the disk already belongs to another disk group and has valid database data. *This data will be lost.* You must be certain that you are selecting a disk that can legitimately be added to the disk group.

   ---

   c. Optionally enter an ASM disk name for each selected disk. (ASM provides a name if you do not.)

8. Click **OK** to create the disk group.

See Also:

- *Oracle Database Storage Administrator's Guide* for a discussion of the `FORCE` option
- *Oracle Database Reference* for information about the various header status types of ASM disks
- "Overview of Disks, Disk Groups, and Failure Groups" on page A-3

Dropping a Disk Group

When you drop a disk group, you delete the disk group, and all of its files. You cannot drop a disk group if any one of its database files is open. After dropping a disk group, you can add its member disks to other disk groups or use them for other purposes.

One reason to drop a disk group is to change redundancy (normal, high, or external). Because you cannot change the redundancy of a disk group, you must drop the disk group and then re-create it with the proper redundancy. In this case, you must back up or move disk group data before you drop the disk group.

To drop a disk group:

1. Go to the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-4.

2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the ASM instance was created.
4. In the **Select** column, select the name of the disk group that you want to drop.
5. Click **Delete**.

   A confirmation page appears.

6. If you want to delete the disk group even if it still contains files, click **Show Advanced Options** and ensure that the **Including Contents** option is selected.

   If the disk group contains files and the **Including Contents** option is not selected, you cannot drop the disk group.

7. On the confirmation page, click **Yes**.

See Also:

- "Overview of Disks, Disk Groups, and Failure Groups" on page A-3

Adding Disks to a Disk Group

You add disks to a disk group to increase the total amount of storage space in a disk group. You can add one or multiple disks in a single operation. ASM then rebalances the disk group so that data is evenly distributed on all disks, including the newly added disks.

You can control the **power** of the rebalance operation, which is a number from 0 to 11. The higher the number, the faster the rebalance operation completes. Lower numbers cause rebalancing to take longer, but use fewer processing and I/O resources. This leaves these resources available for the database. The default value of 1 minimizes disruption to the database. A value of 0 prevents the rebalance operation from happening. Manual or automatic rebalancing can then occur at a later time. For example, you may want to postpone rebalancing because you want to wait for a time when there are fewer demands on the database, or because you want to add more disks or drop disks later and want the rebalancing to be done only once for all disk group changes.

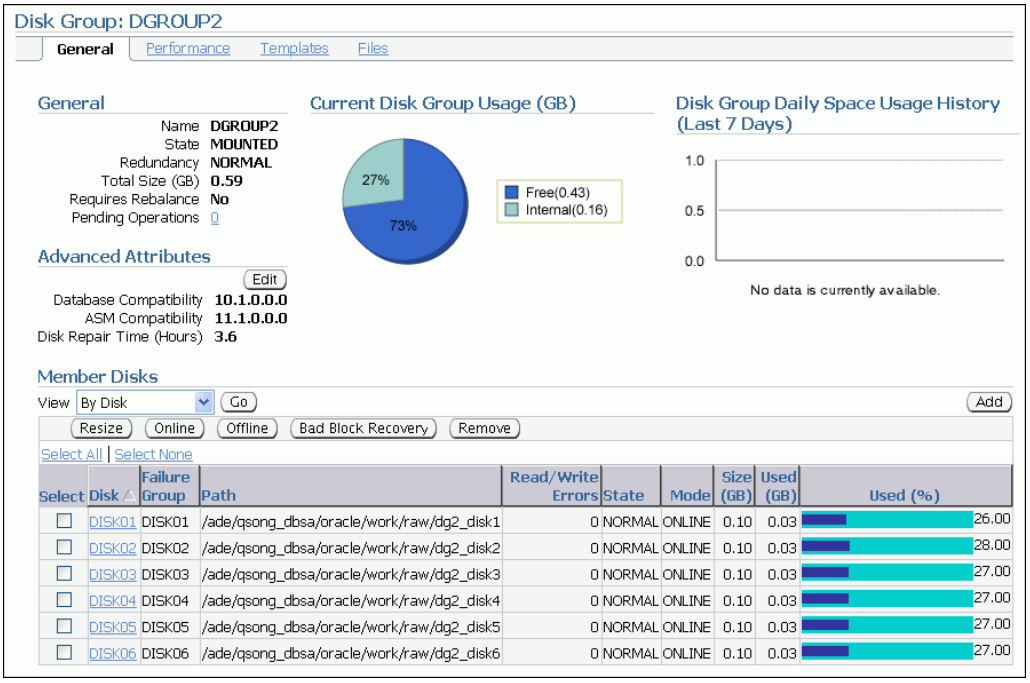
To add one or more disks to a disk group:

1. Go to the ASM Home page.

See "Accessing the Automatic Storage Management Home Page" on page A-4.

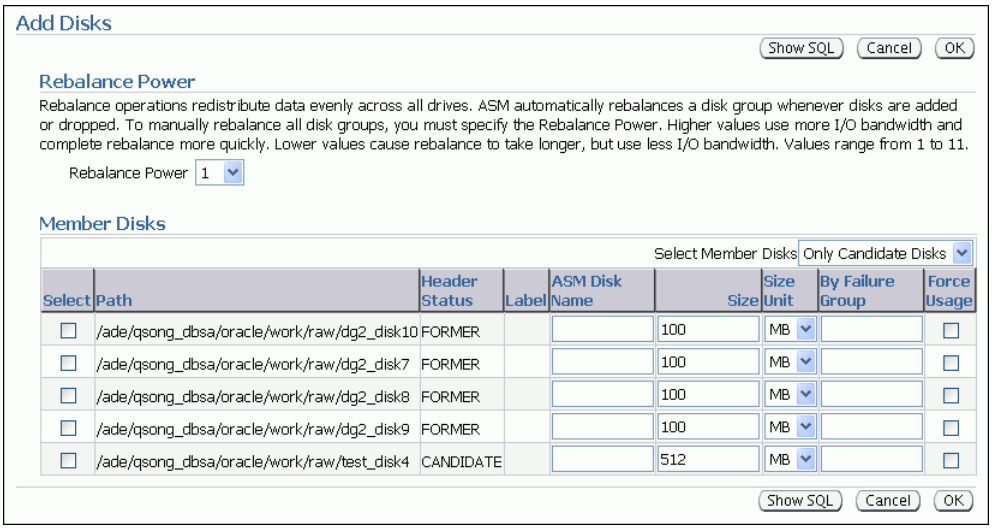
2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the ASM instance was created.
4. Click a link in the **Name** column to select the disk group to which you want to add disks.

   The Disk Group page appears, displaying a list of disks that are already in the disk group.



5. Click **Add**.

   The Add Disks page appears. It displays a list of ASM disks that are available to be added to the disk group. This includes disks with the header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`.



6. If you want to display not only disks that are available to be added to a disk group, but all ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), select **All Disks** from the Select Member Disks list.

   The page is refreshed and now displays the complete list of ASM disks that were discovered by the ASM instance.

7. Optionally change the rebalance power by selecting from the Rebalance Power list.
8. Select disks as follows:
   a. Select the check box to the left of each disk that you want to add to the disk group.
   b. If you want to force the inclusion of a disk in the disk group, select the **Force Usage** check box at the right.

      Caution: The Force Usage check box causes the disk to be added to the disk group even if the disk already belongs to another disk group and has valid database data. *This data will be lost.* You must be certain that you are selecting a disk that can legitimately be added to the disk group.

   c. Optionally enter an ASM disk name for each disk. (ASM provides a name if you do not.)
9. Click **OK** to add the selected disks.

See Also:

- "Overview of Disks, Disk Groups, and Failure Groups" on page A-3
- *Oracle Database Storage Administrator's Guide* for more information about controlling rebalance operations
- *Oracle Database Storage Administrator's Guide* for a discussion of the `FORCE` option

Dropping Disks from a Disk Group

When you drop a disk from the disk group, the disk group is rebalanced by moving all of the file extents from the dropped disk to other disks in the disk group. ASM then releases the disk, and you can then add it to another disk group or use it for other purposes. You can drop one or multiple disks in a single operation. You can also optionally set rebalance power for the drop operation.

The following is a list of possible reasons for dropping a disk:

- A disk is starting to fail, and you need to replace it.
- You want to upgrade a disk.
- You want to reallocate the disk to a different disk group, or reallocate the disk to a different storage system.

Note: Dropping disks from a disk group only logically deletes the disks from the disk group. It does not delete the disk contents. *However, the contents are lost when you add the disk to a new disk group.*

To drop one or more disks from a disk group:

1. Go to the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-4.

2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the ASM instance was created.
4. Click a link in the **Name** column to select the disk group from which you want to drop disks.

   The Disk Group page appears.

5. Under the Member Disks heading, in the **Select** column, select the check boxes for the disks that you want to drop, and then click **Remove**.

   A confirmation page appears.

6. If you want to change rebalance power, or if you want set the `FORCE` option for dropping disks, do the following:

   a. Click **Show Advanced Options**.

   b. Optionally select the **Force** option or select a value for rebalance power in the Rebalance Power list.

   Note: You may need the `FORCE` option if ASM cannot read from or write to the disk.

   ---

7. Click **Yes** to drop the disks.

   The Disk Group page returns, and displays a state of `DROPPING` for each disk being dropped.

8. Refresh the page until the dropped disks no longer appear in the disk group.

   Caution: You cannot reuse or disconnect the dropped disks until the drop and rebalance operations are complete—that is, until the dropped disks no longer appear in the disk group.

   See Also:

   - "Overview of Disks, Disk Groups, and Failure Groups" on page A-3
   - *Oracle Database Storage Administrator's Guide* for more information about the `FORCE` option
   - *Oracle Database Storage Administrator's Guide* for more information and for other caveats about dropping disks

Backing Up ASM-Managed Files

Oracle recommends that you use Recovery Manager (RMAN) to back up files managed by ASM. See Chapter 9, "Performing Backup and Recovery" for the information about using RMAN for performing backups.

See Also:

- "About Automatic Storage Management" on page A-1

Automatic Storage Management: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this appendix and includes annotated screenshots.

To view the Automatic Storage Management OBE, in your browser, enter the following URL:

`http://www.oracle.com/technology/obe/10gr2_2day_dba/asm/asm.htm`

Index

A

adding
  columns to a table, 8-12
  disks to ASM disk groups, A-8
  table constraints, 8-13
ADDM
  See Automatic Database Diagnostics Monitor
administering
  ASM, A-3
  users, 7-9
administrative user accounts, 7-4
  SYS, 5-6, 7-4
  SYSTEM, 5-6
advisors
  Automatic Database Diagnostics Monitor (ADDM), 10-17
  Buffer Cache Advisor, 10-18
  description, 10-16
  Memory Advisor, 10-18
    using, 10-27
  performance, 10-17
  PGA Advisor, 10-18
  repairing critical errors, 11-9
  Segment Advisor, 10-18
    using, 6-19
  SGA Advisor, 10-18
  Shared Pool Advisor, 10-18
  SQL Access Advisor, 10-17
    using, 10-19
  SQL Tuning Advisor, 10-17
    using, 10-24
  Undo Advisor, 6-28, 10-18
alerts
  reviewing critical errors, 11-4
ALTER SYSTEM statement, 5-12
archived redo log files
  advantages of using, 2-11
  description, 2-11, 6-4
ARCHIVELOG mode, 2-11
archiving, 2-11
ARC*n* processes, 5-3
ASM
  See Automatic Storage Management
Automatic Database Diagnostics Monitor (ADDM)
  description, 10-17
  findings, 10-2
  modifying snapshot frequency and retention, 10-16
  overview, 10-2
  snapshots, 10-2
  viewing performance analysis, 10-12
Automatic Diagnostic Repository, 11-1
automatic memory management, 2-13, 5-13
  enabling, 5-15
  modifying settings, 5-16
automatic PGA memory management
  modifying settings, 5-20
automatic shared memory management, 2-13, 5-13
  modifying settings, 5-18
Automatic SQL Tuning Advisor
  about, 10-19
  configuring, 10-20
  viewing results, 10-23
Automatic Storage Management (ASM), 2-4, A-1
  accessing the Home page, A-4
  administering, A-3
  backing up files, A-11
  definition, A-1
  disk discovery, A-3
  failure groups, A-3
  installing, A-4
  mirroring, A-2
  monitoring disk group space usage, A-5
  rebalancing, A-2
Automatic Workload Repository, 10-7
AWR
  See Automatic Workload Repository

B

background processes
  and database configuration, 2-13
  archiver (ARC*n*), 5-3
  checkpoint, 5-3
  database writer (DBW*n*), 5-3
  description, 1-2, 5-3
  log writer (LGWR), 5-3
  manageability monitor (MMON), 5-3
  process monitor (PMON), 4-2, 5-3
  system monitor (SMON), 5-3
  use by database instances, 5-1

backup
available, 9-19
backup pieces, 9-8
channels, 9-8
consistent, 9-2
cross-checking, 9-19, 9-20
deleting expired backups, 9-21
deleting obsolete backups, 9-22
expired, 9-19
files, 6-10
for Automatic Storage Management files, A-11
full backup of datafiles, 9-10
image copies, 9-7
inconsistent, 9-2
incremental backup of datafiles, 9-10
making unavailable, 9-21
managing, 9-18
obsolete, 9-19
Oracle suggested backup strategy, 9-14
overview, 9-1
sets, 9-7
settings, 9-7
tags, 9-11
unavailable, 9-19
validating, 9-16
whole database backup, 9-11
backup and recovery area
specifying during database creation, 2-12
BasicFiles, 8-6
blackouts
defining time periods, 3-8
description, 3-8
block size
configuring for the database, 2-13
Buffer Cache Advisor
description, 10-18

C

central database management
configuring, 2-10
character sets
defining, 2-14
description, 2-14
checkpoint
of the database, 5-3
checkpoint process, 5-3
classes, Java, 8-27
CLOB data type, 8-4
code
database-resident, 8-27
columns
adding, 8-12
deleting, 8-13
comments
adding to problem activity logs, 11-8
compiling schema objects, 8-27
compressed tables, 8-7
configuring
Automatic SQL Tuning Advisor, 10-20
connect descriptors
description, 4-2
example, 4-2
CONNECT role, 7-3
connect strings
description, 4-2
connection modes
configuring, 2-14
connection privileges
SYSASM, A-3
SYSDBA, 5-6
SYSOPER, 5-6
constraints, 8-5
adding, 8-13
deleting, 8-15
modifying, 8-15
control files
description, 6-3
multiplexing, 6-3
SQL*Loader, 8-16
creating
databases
overview, 2-1
steps, 2-5
using standalone DBCA, 2-8
indexes, 8-21
service requests, 11-6
tables, 8-9
users, 7-11
customized repairs, 9-25

D

data
loading, 8-16
data blocks
description of, 2-13, 6-5
data dictionary tables
access to as user SYS, 7-5
data integrity, 8-5
Data Recovery Advisor, 9-22
data types
CLOB, 8-4
DATE, 8-4
NUMBER, 8-4
overview, 8-3
user-defined, 8-6
VARCHAR2, 8-4
database administration
privileges required, 7-4
database checkpoint, 5-3
Database Configuration Assistant (DBCA)
configuring database options, 2-15
deleting a databases, 2-15
managing database templates, 2-15
using to create a database, 2-8
Database Control
See Oracle Enterprise Manager Database Control
database links
description, 8-30

database objects
naming, 8-1
Database Upgrade Assistant (DBUA)
overview, 12-9
restrictions on versions, 12-10
starting, 12-11
steps for upgrading, 12-12
database-resident program code, 8-27
databases
advisors
using, 10-16
alerts, 10-1
responding to alerts, 10-10
setting metric thresholds, 10-8
setting notification rules, 10-11
viewing metrics, 10-7
backup
description, 9-2
backup and recovery, 9-1
configuring options, 2-15
creating with DBCA, 2-8
deleting using DBCA, 2-15
diagnosing performance problems, 10-12
duplicating, 2-16
managing
roadmap, 3-1
with Oracle Enterprise Manager Database Control, 3-1
metric thresholds, 10-1
monitoring
alert-based, 10-1
performance, 10-1
overview of creating, 2-1
performance
advisors, 10-17
monitoring, 10-1
recovery
description, 9-2
point-in-time, 9-3
restoring
description, 9-2
starting and stopping on Microsoft Windows, 5-9
steps for installing and creating, 2-5
storage structure
logical structures, 6-1
physical structures, 6-1
tuning, 10-1
upgrading, 12-9, 12-12
users, 7-1
using DBCA templates, 2-15
datafiles
and the Oracle Suggested Backup Strategy, 9-14
autoextend, 6-8, 6-15
autoextend and Oracle Flashback, 6-26
backing up, 9-2
backups of only used blocks, 9-8
data blocks, 6-5
description of, 6-5
example of corrupt block detection, 9-24
extents, 6-5
full backups, 9-10
image copies of, 9-10
media recovery, 9-3
missing, 9-2
recovering from the loss of, 9-31
recovery without restoration, 9-2
restoring from consistent and inconsistent backups, 9-2
segments, 6-5
SQL*Loader, 8-16
tempfiles, 6-5
validating backups of, 9-17
DATE data type, 8-4
DBA role, 7-4, 8-2
DBCA
See Database Configuration Assistant
DBUA
See Database Upgrade Assistant
DBW*n* processes, 5-3
dedicated server mode, 2-14
dedicated server processes, 5-4
deleting
constraints, 8-15
databases, 2-15
table columns, 8-13
tables, 8-18
users, 7-18
views, 8-26
disk discovery
in Automatic Storage Management, A-3
disk groups
adding disks to, A-8
creating, A-6
dropping, A-8
dropping disks from, A-10
monitoring disk space usage, A-5
overview, A-3
dropping
disk groups, A-8
disks from disk groups, A-10
tablespaces, 6-18
undoing with Flashback Drop, 9-28
duplicating
databases, 2-16
users, 7-12

E

emctl, 3-3
enabling
automatic memory management, 5-15
Enterprise Manager
See Oracle Enterprise Manager Database Control
EXAMPLE tablespace, 6-6
configuring, 2-12
expiring passwords
reasons for, 7-17
extending
tablespaces, 6-8, 6-15
tablespaces and Oracle Flashback, 6-26

undo tablespaces, 6-25, 6-26
extents
description of, 6-5

F

failure groups, A-3
flash recovery area, 9-3
configuring, 9-6
retention policies, 9-5
specifying during database creation, 2-12
Flashback Database, 9-30
Flashback Drop, 9-28
flashback features
Flashback Database, 9-30
Flashback Drop, 9-28
Flashback Table, 9-26
Flashback Table, 9-26, 9-27
functions, PL/SQL, 8-27

G

global database names, 2-3
global indexes, 8-20
granting
privileges, 7-13
roles, 7-13
grid control
configuring, 2-10

I

incidents
about, 11-2
closed, viewing, 11-10
closing, 11-9
tracking, 11-2
indexes
creating, 8-21
description, 8-18
global, 8-20
partitioned, 8-20
viewing, 8-20
initialization parameters
description, 5-2
server parameter files, 5-2
viewing and modifying, 5-10
installation
checking prerequisites, 2-2
choices, 2-2
of ASM, A-4
overview, 2-1
steps, 2-5
storage options, 2-4
using DBCA for database, 2-8
instances
database instances, 5-1
management, 5-1
memory structure, 5-4
Oracle instances, 5-1
PGA
definition, 5-6
shutdown, 5-7
startup, 5-6
invalid schema objects, 8-27

J

Java classes, 8-27
Java source code, 8-27

L

LGWR process, 5-3
listener.ora files, 4-2
listeners
See Oracle listeners
Load Data wizard
using, 8-17
loading data, 8-16
LOB columns, 8-6
local database management
configuring, 2-10
locking and unlocking users, 7-17
log switch
description, 2-11

M

managing
memory, 5-12
media recovery, 9-3
memory
configuring, 2-13
management, 5-4, 5-12, 5-13
requirements, 2-2
structure of in an Oracle instance, 5-4
Memory Advisors
description, 10-18
using, 10-27
mirroring, A-2
MMON process, 5-3
modifying
initialization parameters, 5-10
table attributes, 8-12
table constraints, 8-15
users, 7-16

N

Net Configuration Assistant (NETCA)
description, 4-3
NETCA
See Net Configuration Assistant (NETCA)
network configuration
connection requests, 4-1
description, 4-1
notification of alerts
alert notification, 3-8
NUMBER data type, 8-4

O

object privileges, 7-3
offline tablespaces, 6-17
Oracle Enterprise Manager Database Control
  accessing the ASM Home page, A-4
  backup, 9-11
  backup and recovery, 9-4
  configuration information, 2-8
  description, 3-2
  features, 3-2
  granting administrative privileges to use, 3-6
  managing databases, 3-1
  online help, 3-3
  starting and stopping, 3-3
Oracle listeners
  listener.ora file, 4-2
  starting
Oracle Net
  description, 4-1
  listener configuration, 4-1
  mapping methods, 4-3
    directory naming, 4-3
    easy connect naming, 4-3
    local naming, 4-3
Oracle suggested repair, 9-22
Oracle system identifier, 2-3
ORACLE_BASE environment variable, 2-6, 2-7
ORACLE_HOME environment variable, 2-6, 4-6

P

partitioned indexes, 8-20
partitioned tables, 8-6
password files, 6-9
password policies
  default, 7-19
  description of, 7-19
  setting, 7-18
passwords
  expiring, 7-17
patch sets, 12-1
patches, 12-1
Patching wizard, 12-5
PGA
  See Program Global Area
PGA Advisor
  description, 10-18
PMON process, 4-2, 5-3
predefined roles, 7-3
privileges
  administrative, 7-4
  and synonyms, 8-30
  authenticating SYSDBA and SYSOPER users, 6-9
  connection, 5-6
  for installation of Oracle Database sofware, 2-5
  granting, 7-13
  granting to a Database Control administrative user, 3-7
  object, 7-3
  recommended, 7-1
  required for database administration, 3-6, 7-4
  revoking a system, 7-14
  schema object management, 8-2
  SYSASM, A-3
  SYSDBA, 5-6, 9-4
  SYSOPER, 5-6
  system, 7-3
  taking tablespaces offline, 6-17
  users, 7-3
    using roles to manage, 7-3
problem activity logs
  adding comments to, 11-8
problems
  about, 11-1
  adding comments to activity logs, 11-8
  problem key, 11-1
  viewing, 11-9
  viewing detail, 11-5
procedures, 8-27
profiles, 7-19
Program Global Area (PGA)
  components, 5-6
  definition, 5-6
  description, 5-5
  target setting, 10-27

Q

quotas
  assigning for a tablespace, 7-15

R

raw devices, 2-4
rebalancing, A-2
recovery
  complete, 9-3
  configuration, 2-11
  flash recovery area, 9-3
  incomplete, 9-3
  media, 9-3
  overview, 9-1
  whole database recovery, 9-30, 9-31
Recovery Manager (RMAN)
  channels, 9-8
  control file use, 9-4
  overview, 9-1
  parallelism, 9-8
  recovery catalog, 9-4
  repository, 9-4
recycle bins, 8-18
redo log files
  description, 2-11, 6-3
  multiplexing, 6-3, 6-23
redo log groups
  description, 2-11
reference collection, 10-5
repairs
  customized, 9-25
  Oracle suggested, 9-22

RESOURCE role, 7-3, 8-2
revalidating schema objects, 8-27
revoking
  roles, 7-14
  system privileges, 7-14
roles
  administering, 7-5
  CONNECT, 7-3
  DBA, 7-4, 8-2
  granting, 7-13
  predefined, 7-3
  RESOURCE, 7-3, 8-2
  revoking, 7-14
  using to manage user privileges, 7-3
  viewing, 7-6
rollback segments
  description, 6-5

S

sample schemas
  configuring, 2-12
  installation of, 2-3
schema objects
  compiling, 8-27
  database links, 8-30
  description of, 8-1
  indexes, 8-18
    creating, 8-21
    viewing, 8-20
  invalid, 8-27
  privileges, 8-2
  sequences, 8-29
  synonyms, 8-29
  tables, 8-3
    constraints, 8-5
    modifying, 8-12
  validating, 8-27
  views, 8-23
schemas, 7-2
SecureFiles, 8-6
Segment Advisor
  description, 10-18
  using, 6-19
segments
  description of, 6-5
sequences
  description, 8-29
server parameter files, 5-2
server processes
  description, 5-4
service requests
  creating, 11-6
  tracking, 11-8
services
  on Microsoft Windows, 5-9
session
  description of, 5-5
setting
  password policies, 7-18
SGA
  See System Global Area
Shared Pool Advisor
  description, 10-18
shared server mode, 2-14
shared server processes, 5-4
shutting down databases
  Microsoft Windows, 5-9
SID, 2-3
SMON process, 5-3
snapshots, 10-2
source code, Java, 8-27
SQL Access Advisor
  using, 10-19
SQL Advisors
  about, 10-18
  description, 10-17
SQL Profile, 10-20
SQL Response Time, 10-5
SQL statements
  about, 3-11
SQL Tuning Advisor
  about, 10-18
  using, 10-24
SQL Tuning Set, 10-5
SQL*Loader
  control file, 8-16
starting databases
  Microsoft Windows, 5-9
startup and shutdown
  about, 5-6
  privileges required for, 5-6
subprograms, 8-27
Support Workbench
  about, 11-1
  Custom Packaging process, 11-7
  home page, 11-2
  Quick Packaging process, 11-7
  viewing problems, 11-9
synonyms
  and privileges, 8-30
  description, 8-29
SYS user, 5-6, 7-4
SYSASM system privilege, A-3
SYSAUX tablespace, 6-6
SYSDBA system privilege, 5-6, 7-5
  authentication, 6-9
SYSOPER system privilege, 7-5
  authentication, 6-9
System Global Area (SGA)
  components, 5-5
  description, 5-5
  target setting, 10-27
system identifier (SID), 2-3
system privileges, 7-3
  SYSASM, A-3
  SYSDBA and SYSOPER, 7-5
SYSTEM tablespace, 6-6
SYSTEM user, 5-6, 7-4

T

tables
  adding columns to, 8-12
  adding constraints, 8-13
  compressed, 8-7
  creating, 8-9
  deleting, 8-18
  deleting columns, 8-13
  deleting constraints, 8-15
  description, 8-3
  loading data into, 8-16
  LOB columns, 8-6
  modifying attributes, 8-12
  modifying constraints, 8-15
  partitions, 8-6
  retrieving a dropped table, 9-29
  viewing data in, 8-8
  viewing information about, 8-7
tablespaces
  autoextend, 6-8, 6-15
  autoextend and Oracle Flashback, 6-26
  creating, 6-12
  description, 6-6
  dictionary-managed, 6-7
  dropping, 6-18
  EXAMPLE, 6-6
  extending undo tablespaces, 6-25, 6-26
  locally managed, 6-7
  modifying, 6-15
  offline, 6-8, 6-17
  quotas, 7-15
  read only, 6-8
  read write, 6-8
  Segment Advisor, 6-15
  SYSAUX, 6-6
  SYSTEM, 6-6
  TEMP, 6-7
  temporary, 6-8
  types of, 6-7
  UNDOTBS1, 6-7
  USERS, 6-7
TEMP tablespace, 6-7
templates
  creating databases with, 2-10
  using with DBCA, 2-15
triggers, 8-27

U

undo
  about, 6-25
  advisor, 6-28
  managing, 6-25, 6-26
  tablespaces, 6-7
  viewing, 6-27
Undo Advisor
  computing minimum undo tablespace size with, 6-28
  description, 10-18
  setting minimum undo retention period with, 6-28
undo retention period
  setting with Undo Advisor, 6-28
undo tablespaces
  autoextend, 6-25, 6-26
  changing to fixed-size, 6-30
  computing minimum size for fixed-size tablespace, 6-28
  description of, 6-7, 6-25
UNDOTBS1 tablespace, 6-7
Unicode
  description, 2-14
upgrading a database, 12-12
user-defined data types, 8-6
users
  accounts
    administrative, 7-4
    description, 7-1
  administering, 7-9
  creating, 7-11
  creating by duplication, 7-12
  deleting, 7-18
  duplicating, 7-12
  expiring passwords for, 7-17
  locking and unlocking, 7-17
  modifying, 7-16
  privileges, 7-3
  SYSASM, A-3
  viewing, 7-10
USERS tablespace, 6-7

V

validating schema objects, 8-27
VARCHAR2 data type, 8-4
viewing
  indexes, 8-20
  initialization parameters, 5-10
  table data, 8-8
  table information, 8-7
  users, 7-10
  views, 8-24
views
  deleting, 8-26
  description, 8-23
  displaying, 8-24

W

wizards
  ASM migration, A-2
  Create Blackout, 3-9
  database creation, 2-9
  incident reporting, 11-3, 11-7
  Load Data wizard, 8-16
  Patching, 12-5
  Quick Packaging, 11-7
  Recovery, 9-3, 9-25, 9-30
  Schedule Customized Backup, 9-15
  SQL Access Advisor, 10-26